

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **07182165 A**(43) Date of publication of application: **21 . 07 . 95**

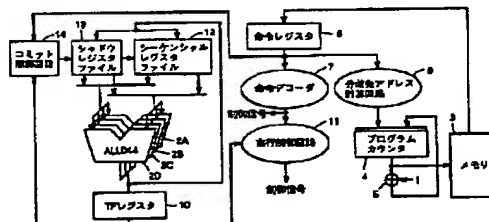
(51) Int. Cl.

G06F 9/38
G06F 9/32
(21) Application number: **06133106**(22) Date of filing: **15 . 06 . 94**
(30) Priority: **28 . 09 . 93 JP 05241162**
12 . 11 . 93 JP 05283521
(71) Applicant: **MITSUBISHI ELECTRIC CORP**(72) Inventor: **ANDO HIDEKI**
**(54) PROCESSING METHOD/DEVICE FOR
INSTRUCTION WITH COMMITTING CONDITION**
(57) Abstract:

PURPOSE: To carry out an instruction to be carried out after a conditional instruction before execution of this conditional instruction.

CONSTITUTION: This method/device is provided with a normal sequential register file 12, a shadow register file 13 which holds the result data when an instruction to be carried out after a conditional instruction is carried out before the conditional instruction, and a TF register 10 which holds the information showing whether the relevant condition is true, false or undecided. Furthermore a commitment control circuit 14 is added to hold the condition corresponding to the result data stored in the file 13 and to transfer the result data to the file 12 when the coincidence is secured between the truth/falsehood of the condition and that of the register 10.

COPYRIGHT: (C)1995,JPO



X-11

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 7-182165

(43) 公開日 平成7年(1995)7月21日

(51) Int. Cl.^o 識別記号 庁内整理番号 FI 技術表示箇所
G 0 6 F 9/38 3 3 0 J
9/32 3 2 0 F

審査請求 未請求 請求項の数 17 OL

(全 34 頁)

(21) 出願番号 特願平 6-133106

(22) 出願日 平成6年(1994)6月15日

(31) 優先権主張番号 特願平 5-241162

(32) 優先日 平 5 (1993) 9月28日

(33) 優先権主張国 日本 (JP)

(31) 優先権主張番号 特願平 5-283521

(32) 優先日 平 5 (1993) 11月12日

(33) 優先権主張国 日本 (JP)

(71) 出願人 00006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 安藤 秀樹

兵庫県伊丹市瑞原4丁目1番地 三菱電機株

式会社システムエル・エス・アイ開発研究

所内

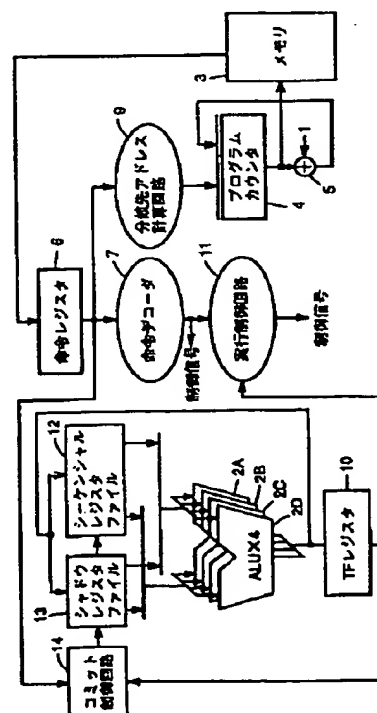
(74) 代理人 弁理士 深見 久郎 (外3名)

(54) 【発明の名称】 コミット条件付き命令の処理方法およびその装置

(57) 【要約】

【目的】 条件命令を実行した後に実行すべき命令をその条件命令を実行する前に実行できるようにする。

【構成】 このマイクロプロセッサは、通常のシーケンシャルレジスタファイル12と、条件命令を実行した後に実行すべき命令をその条件命令を実行する前に実行した場合の結果データを保持するシャドウレジスタファイル13と、その条件が「真」であるか、「偽」であるか、または「未定」であるかの情報を保持するTFレジスタ10と、シャドウレジスタファイル13に格納した結果データに対応する条件を保持し、その条件の真偽とTFレジスタ10における真偽とが一致した時点でそのシャドウレジスタファイルに格納されている結果データをシーケンシャルレジスタファイル12へ転送するコミット制御回路14とを備える。



【特許請求の範囲】

【請求項1】 複数の条件の下で、各々が対応する1つのコミット条件を持つ複数の命令を処理するコミット条件付き命令の処理方法であって、前記コミット条件は対応する命令を最終的に処理するために必要な前記条件のうちいずれかからなり、

前記条件のうち選択された少なくとも1つの条件を判定するステップと、

前記命令のうち選択された少なくとも1つの命令に対応するコミット条件を前記条件と比較するステップと、

その比較後、その選択された命令を実行して結果データを生成するステップと、

その対応するコミット条件中のすべての条件の真偽がすでに判定されている条件の真偽と対応的に一致している第1の場合はその生成した結果データを確定的に保持し、その対応するコミット条件中のいずれかの条件の真偽がすでに判定されている条件の真偽と対応的に一致しかつその対応するコミット条件中のその他の条件が未だ判定されていない第2の場合はその生成した結果データを暫定的に保持するステップと、

前記第2の場合は、その対応するコミット条件中のすべての条件の真偽がすでに判定されている条件の真偽と対応的に一致するコミット時に、その暫定的に保持した結果データを確定的に保持し直すステップとを含むコミット条件付き命令の処理方法。

【請求項2】 その選択された命令が処理される間に例外が発生した場合であって、前記第1の場合はその例外を直ちに処理し、前記第2の場合はその例外の発生を記憶するステップと、

その例外の発生が記憶されている場合は前記コミット時にその例外を処理するステップとをさらに含む請求項1に記載のコミット条件付き命令の処理方法。

【請求項3】 その選択された命令が処理される間に例外が発生した場合であって、前記第1の場合はその例外を直ちに処理し、前記第2の場合はその例外の発生を記憶するステップと、

前記コミット時にその例外の発生が記憶されている場合は前記複数の命令のうち前記コミット時まですでに処理された命令を最初から再処理するステップと、

その命令が再処理される間にその例外が再び発生したとき、その例外を直ちに処理するステップとをさらに含むことを特徴とする請求項1に記載のコミット条件付き命令の処理方法。

【請求項4】 複数の条件の下で、各々が対応する1つのコミット条件を持つ複数の命令を複数の処理ステージを経てパイプライン制御によって処理するコミット条件付き命令の処理方法であって、前記コミット条件は対応する命令を最終的に処理するために必要な前記条件のうちいずれかからなり、

前記条件のうち選択された少なくとも1つの条件を判定

するステップと、

前記複数の処理ステージのうち1つの処理ステージにおいて、前記命令のうち選択された少なくとも1つの命令に対応するコミット条件を前記条件と比較するステップと、

その対応するコミット条件中のすべての条件の真偽がすでに判定されている条件の真偽と対応的に一致している第1の場合はその次の処理ステージへ移行するときにその選択された命令が実行されて生成されるであろう結果データは確定的に保持されるべきという第1の処置情報を生成し、その対応するコミット条件中のいずれかの条件の真偽がすでに判定されている条件の真偽と対応的に一致しかつその対応するコミット条件中のその他の条件が未だ判定されていない第2の場合はその次の処理ステージへ移行するときにその選択された命令が実行されて生成されるであろう結果データは暫定的に保持されるべきという第2の処置情報を生成するステップと、

その比較後、その選択された命令を実行して結果データを生成するステップと、

前記第1の場合はその第1の処置情報に従ってその生成された結果データを確定的に保持し、前記第2の場合はその第2の処置情報に従ってその生成された結果データを暫定的に保持するステップと、

前記第2の場合は、その対応するコミット条件中のすべての条件の真偽がすでに判定されている条件の真偽と対応的に一致するコミット時に、その暫定的に保持した結果データを確定的に保持し直すステップとを含むコミット条件付き命令の処理方法。

【請求項5】 第1または第2の処置情報を生成した後、前記1つの処理ステージ以降のもう1つの処理ステージにおいて、その対応するコミット条件を前記条件と再び比較するステップと、

その比較の結果、前記第2の場合から前記第1の場合へ変化している場合はさらにその次の処理ステージへ移行するときにその第2の処置情報を第1の処置情報へ変更するステップとをさらに含むことを特徴とする請求項4に記載のコミット条件付き命令の処理方法。

【請求項6】 複数の条件の下で、各々が対応する1つのコミット条件を持つ複数の命令を処理するコミット条件付き命令の処理装置であって、前記コミット条件は対応する命令を最終的に処理するために必要な前記条件のうちいずれかからなり、

前記命令のうち選択された少なくとも1つの命令を実行し、かつ、前記条件のうち選択された少なくとも1つの条件を判定する演算手段と、

前記演算手段がその選択された命令を実行して生成した結果データを確定的または暫定的に保持するデータ保持手段と、

前記複数の条件に対応する複数のエントリを含み、前記エントリの各々は、対応する条件が真であるという第1

の条件情報、対応する条件が偽であるという第2の条件情報、および対応する条件の真偽が未定であるという第3の条件情報のうちいずれかを保持するものである、真偽レジスタ手段と、

その対応するコミット条件を前記真偽レジスタ手段から得られた条件情報と比較し、(i) その対応するコミット条件中のすべての条件の真偽がそのすべての条件に対応する条件情報と対応的に一致する第1の場合は前記データ保持手段が結果データを確定的に保持するように制御し、(ii) その対応するコミット条件中のいずれかの条件の真偽がそのいずれかの条件に対応する条件情報と対応的に一致しかつその対応するコミット条件中のその他の条件に対応する条件情報が第3の条件情報である第2の場合は前記データ保持手段が結果データを暫定的に保持するように制御する実行制御手段と、

その対応するコミット条件を前記真偽レジスタ手段から得られた条件情報と比較し、その対応するコミット条件中のすべての条件の真偽がそのすべての条件に対応する条件情報と対応的に一致するコミット時に、前記データ保持手段が暫定的に保持した結果データを確定的に保持し直すように制御するコミット制御手段とを備えたコミット条件付き命令の処理装置。

【請求項7】 複数の条件の下で、各々が対応する1つのコミット条件を持つ複数の命令を複数の処理ステージを経てパイプライン制御によって処理するコミット条件付き命令の処理装置であって、前記コミット条件は対応する命令を最終的に処理するために必要な前記条件のうちいずれかからなり、

前記命令のうち選択された少なくとも1つの命令を実行し、かつ、前記条件のうち選択された少なくとも1つの条件を判定する演算手段と、

前記演算手段がその選択された命令を実行して生成した結果データを確定的または暫定的に保持するデータ保持手段と、

前記複数の条件に対応する複数のエントリを含み、前記エントリの各々は、対応する条件が真であるという第1の条件情報、対応する条件が偽であるという第2の条件情報、および対応する条件の真偽が未定であるという第3の条件情報のうちいずれかを保持するものである、真偽レジスタ手段と、

前記複数の処理ステージのうち1つの処理ステージにおいて、その対応するコミット条件を前記真偽レジスタ手段から得られた条件情報と比較し、(i) その対応するコミット条件中のすべての条件の真偽がそのすべての条件に対応する条件情報と対応的に一致する第1の場合はその次の処理ステージへ移行するときに前記演算手段がその選択された命令を実行して生成するであろう結果データは確定的なものであるという第1の処置情報を生成し、(ii) その対応するコミット条件中のいずれかの条件の真偽がそのいずれかの条件に対応する条件情報と対

応的に一致しかつその対応するコミット条件中のその他の条件に対応する条件情報が第3の条件情報である第2の場合はその次の処理ステージへ移行するときに前記演算手段がその選択された命令を実行して生成するであろう結果データは暫定的なものであるという第2の処置情報を生成する情報生成手段と、

(i) 前記第1の場合はその第1の処置情報に従って前記演算手段が生成した結果データを前記データ保持手段が確定的に保持するように制御し、(ii) 前記第2の場合はその第2の処置情報に従って前記演算手段が生成した結果データを前記データ保持手段が暫定的に保持するように制御する書込制御手段と、

その対応するコミット条件を前記真偽レジスタ手段から得られた条件情報と比較し、その対応するコミット条件中のすべての条件の真偽がそのすべての条件に対応する条件情報と対応的に一致するコミット時に、前記データ保持手段が暫定的に保持した結果データを確定的に保持し直すように制御するコミット制御手段とを備えたコミット条件付き命令の処理装置。

【請求項8】 前記情報生成手段が第1または第2の処置情報を生成した前記1つの処理ステージ以降のもう1つの処理ステージにおいて、前記第2の場合から前記第1の場合へ変化している場合はさらにその次の処理ステージへ移行するときにその第2の処置情報を第1の処置情報へ変更する情報変更手段をさらに備えた請求項7に記載のコミット条件付き命令の処理装置。

【請求項9】 前記データ保持手段は、その結果データを確定的に保持する第1のレジスタ手段と、その結果データを暫定的に保持する第2のレジスタ手段とを備えたことを特徴とする請求項6から請求項8までのいずれかに記載のコミット条件付き命令の処理装置。

【請求項10】 前記データ保持手段は、その結果データを確定的に保持しているのかまたは暫定的に保持しているのかを特定するためのフラグを備えたことを特徴とする請求項6から請求項8までのいずれかに記載のコミット条件付き命令の処理装置。

【請求項11】 前記条件のうち少なくとも1つの条件を判定するための条件命令もまた、対応する1つのコミット条件を持ち、

前記真偽レジスタ手段の各エントリは、条件情報を確定的または暫定的に保持し、

前記実行制御手段は、その条件命令に対応するコミット条件を前記真偽レジスタ手段から得られた条件情報と比較し、(i) その条件命令に対応するコミット条件中のすべての条件の真偽がそのすべての条件に対応する条件情報と対応的に一致する第3の場合は前記演算手段がその条件命令に従ってその選択された条件を判定して生成した第1または第2の条件情報を前記真偽レジスタ手段が確定的に保持するように制御し、(ii) その条件命令に対応するコミット条件中のいずれかの条件の真偽がそ

のいずれかの条件に対応する条件情報と対応的に一致しかつその条件命令に対応するコミット条件中のその他の条件に対応する条件情報が第3の条件情報である第4の場合は前記演算手段がその条件命令に従ってその選択された条件を判定して生成した第1または第2の条件情報を前記真偽レジスタ手段が暫定的に保持するように制御し、

その条件命令に対応するコミット条件を前記真偽レジスタ手段から得られた条件情報と比較し、その条件命令に対応するコミット条件中のすべての条件の真偽がそのすべての条件に対応する条件情報と一致するコミット時に、前記真偽レジスタ手段が暫定的に保持した第1または第2の条件情報を確定的に保持し直すように制御する第2のコミット制御手段をさらに備えたことを特徴とする請求項6から請求項10までのいずれかに記載のコミット条件付き命令の処理装置。

【請求項12】 その選択された命令が処理される間に例外が発生しかつ前記第2の場合はその例外の発生を記憶する例外記憶手段と、

前記コミット時に前記例外記憶手段がその例外の発生を記憶している場合はその例外を処理する例外処理手段とをさらに備えたことを特徴とする請求項6から請求項11までのいずれかに記載のコミット条件付き命令の処理装置。

【請求項13】 前記例外処理手段はさらに、その例外を処理する前に、前記複数の命令のうち、少なくともその例外の発生時から前記コミット時までの間にその結果データが暫定的に保持されている命令を再処理する再処理手段を備えたことを特徴とする請求項12に記載のコミット条件付き命令の処理装置。

【請求項14】 前記再処理手段は、その対応する結果データが暫定的に保持されている命令を逐次的な順序で再処理することを特徴とする請求項13に記載のコミット条件付き命令の処理装置。

【請求項15】 前記真偽レジスタ手段から得られた条件情報に前記演算手段がその選択された条件を判定して生成した条件情報を加える真偽計算手段と、

その選択された命令が処理される間に例外が発生しかつ前記第2の場合はその例外の発生を記憶する例外記憶手段と、

前記真偽計算手段から得られた条件情報を保持する将来真偽レジスタ手段と、

前記コミット時に前記例外記憶手段がその例外の発生を記憶している場合は前記複数の命令のうち前記コミット時まですでに処理された命令を最初から再処理する再処理手段と、

前記再処理手段がその命令を再処理している間にその例外が再び発生したとき、その例外を引起こした命令に対応するコミット条件を前記将来真偽レジスタ手段から得られた条件情報と比較し、その対応するコミット条件中

のすべての条件の真偽がそのすべての条件に対応する条件情報と対応的に一致する場合はその例外を処理する例外処理手段とをさらに備え、

前記真偽レジスタ手段は、前記再処理手段がその命令を再処理している場合は前記真偽計算手段から得られた条件情報を受入れず、そうでない場合はその条件情報を受入れ、かつ、前記再処理手段がその命令を再処理し終えたとき前記将来真偽レジスタ手段から得られた条件情報を受入れることを特徴とする請求項6から請求項11までのいずれかに記載のコミット条件付き命令の処理装置。

【請求項16】 前記再処理手段は、その命令のうち前記第2の場合における命令を選択的に再処理すること特徴とする請求項15に記載のコミット条件付き命令の処理装置。

【請求項17】 前記例外記憶手段がその例外の発生を記憶している間はその記憶されている内容が書換えられるのを禁止する禁止手段をさらに備えたことを特徴とする請求項12から請求項16までのいずれかに記載のコミット条件付き命令の処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明はコミット条件付き命令の処理方法およびその装置に関し、さらに詳しくは、複数の条件の下で、各々が対応する1つのコミット条件を持つ複数の命令を処理する方法およびその装置に関する。ここで、コミット条件は対応する命令を最終的に処理するために必要ないずれかの条件から構成される。

【0002】

30 【従来の技術】 図28は、従来の命令処理装置であるマイクロプロセッサの構成を示すブロック図である。

【0003】 図28を参照して、このマイクロプロセッサは、複数のレジスタからなるレジスタファイル1と、演算論理装置（ALU）2と、命令を保持するメモリ3と、実行されるべき命令のアドレスを保持し、かつそのアドレスをメモリ3へ与えるプログラムカウンタ4と、プログラムカウンタ4をインクリメントする回路5とを備える。

40 【0004】 このマイクロプロセッサはさらに、メモリ3からフェッチされた命令を保持する命令レジスタ6と、その命令を解読し、レジスタファイル1の読出／書込を制御するための制御信号およびALU2を制御するための制御信号を生成する命令デコーダ7と、レジスタファイル1から読出されたデータに従って分岐条件を判定するテスト回路8と、テスト回路8からの制御信号にตอบสนองして分岐先アドレスを計算する回路9とを備える。

50 【0005】 このマイクロプロセッサによれば、まずプログラムカウンタ4によってメモリ3に保持されている命令の中から実行すべき命令のアドレスが指定される。その指定された命令はメモリ3からフェッチされ、命令

レジスタ6に保持される。命令レジスタ6によって保持された命令は命令デコーダ7によって解読され、その命令の内容に応答してレジスタファイル1の読出/書込が制御されるとともに、ALU2が制御される。

【0006】命令レジスタ6にフェッチされた命令が演算命令の場合、レジスタファイル1の中の2つのレジスタからデータがそれぞれ読出される。それら読出されたデータはALU2によって演算され、その結果データは再びレジスタファイル1の中のレジスタに書込まれる。

【0007】一方、命令レジスタ6にフェッチされた命令が分岐命令の場合、レジスタファイル1から読出された2つのデータはテスト回路8に与えられる。それらデータに従ってテスト回路8はその分岐条件は満たされているか否かを判定する。すなわち、テスト回路8によってその条件が真であるか偽であるかが判断される。

【0008】そして、その条件が真であれば、分岐先アドレス計算回路9によってその分岐先アドレスが計算され、プログラムカウンタ4に与えられる。その条件が偽であれば、プログラムカウンタ4はインクリメントされる。

【0009】一般に、マイクロプロセッサは、命令を実行している最中に例外と呼ばれる処理を行なわなければならない場合がある。たとえば仮想記憶のマイクロプロセッサにおいて、メモリ3からレジスタファイル1中のレジスタヘデータをフェッチする命令（ロード命令）が実行されるとき、そのロードしようとするデータがメモリ3上に存在しない場合がある。このような例外は一般に「ページフォールト」と呼ばれる。この場合、例外ハンドラと呼ばれる特別なルーチンを呼出してディスクからメモリ3へデータをロードする必要がある。また、その後、例外を引き起こした命令から再び処理を開始する必要がある。

【0010】

【発明が解決しようとする課題】図10は、プログラムの一例を示すフローチャートである。このフローチャートによれば、基本ブロックK1における条件C1: $if (R2 < R3)$ が判定された後、基本ブロックL1およびL3のいずれかが開始される。すなわち、基本ブロックL1における命令I2: $r4 = r12 + r13$ および命令I3: $r5 = r12 - r13$ は、条件C1の下で実行されるべきものである。

【0011】上記従来のマイクロプロセッサによれば、条件C1が判定された後でなければ、命令I2およびI3のいずれも実行され得ない。たとえば、命令I2が実行された結果データがレジスタファイル1におけるレジスタr4に書込まれるのは、条件C1が真であるときだけである。もし条件C1が偽であるのに命令I2が実行された結果データがレジスタr4に書込まれると、レジスタr4が持っていた前のデータが破壊されるからである。

【0012】このように、従来のマイクロプロセッサは、条件C1およびC2の下で実行すべき命令I2、I3およびI4を、条件C1およびC2を判定する前に実行することはできなかった。

【0013】近年、複数のALUを持つマイクロプロセッサが提供されているが、このようなマイクロプロセッサであっても1または2以上の条件下で実行すべき命令を、それら条件を判定する前に実行することはできない。そのため、それら複数のALUのうちいくつかは全く演算をしていないときがあり、演算処理の十分な高速化は実現され得ない。

【0014】この発明の目的は、条件判定後に実行すべき命令を、条件判定前に実行することができるマイクロプロセッサを提供することである。

【0015】この発明の他の目的は、より高速に命令を処理することができるマイクロプロセッサを提供することである。

【0016】この発明のさらに他の目的は、例外が発生したときも高速に命令を処理することができるマイクロプロセッサを提供することである。

【0017】

【課題を解決するための手段】請求項1に記載のコミット条件付き命令の処理方法は、複数の条件の下で、各々が対応する1つのコミット条件を持つ複数の命令を処理することを前提とする。ここで、コミット条件は、対応する命令を最終的に処理するために必要な上記条件のうちいずれかからなる。上記処理方法は、上記条件のうち選択された少なくとも1つの条件を判定するステップと、上記命令のうち選択された少なくとも1つの命令に対応するコミット条件を上記条件と比較するステップと、その比較後、その選択された命令を実行して結果データを生成するステップと、その対応するコミット条件中のすべての条件の真偽がすでに判定されている条件の真偽と対応的に一致している第1の場合はその生成した結果データを確定的に保持し、その対応するコミット条件中のいずれかの条件の真偽がすでに判定されている条件の真偽と対応的に一致しかつその対応するコミット条件中のその他の条件が未だ判定されていない第2の場合はその生成した結果データを暫定的に保持するステップと、上記第2の場合は、その対応するコミット条件中のすべての条件の真偽がすでに判定されている条件の真偽と対応的に一致するコミット時に、その暫定的に保持した結果データを確定的に保持し直すステップとを含む。

【0018】請求項2に記載のコミット条件付き命令の処理方法は、上記請求項1に記載の処理方法に加えてさらに、その選択された命令が処理される間に例外が発生した場合であって、上記第1の場合はその例外を直ちに処理し、上記第2の場合はその例外の発生を記憶するステップと、その例外の発生が記憶されている場合は上記コミット時にその例外を処理するステップとを含む。

【0019】請求項3に記載のコミット条件付き命令の処理方法は、上記請求項1に記載の処理方法に加えてさらに、その選択された命令が処理される間に例外が発生した場合であって、上記第1の場合はその例外を直ちに処理し、上記第2の場合はその例外の発生を記憶するステップと、上記コミット時にその例外の発生が記憶されている場合は上記複数の命令のうち上記コミット時まですでに処理された命令を最初から再処理するステップと、その命令が再処理される間にその例外が再び発生したとき、その例外を直ちに処理するステップとを含む。

【0020】請求項4に記載のコミット条件付き命令の処理方法は、複数の条件の下で、各々が対応する1つのコミット条件を持つ複数の命令を複数の処理ステージを経てパイプライン制御によって処理することを前提とする。ここで、上記コミット条件は、対応する命令を最終的に処理するために必要な上記条件のうちいずれかからなる。上記処理方法は、上記条件のうち選択された少なくとも1つの条件を判定するステップと、上記複数の処理ステージのうち1つの処理ステージにおいて、上記命令のうち選択された少なくとも1つの命令に対応するコミット条件を上記条件と比較するステップと、その対応するコミット条件中のすべての条件の真偽がすでに判定されている条件の真偽と対応的に一致している第1の場合はその次の処理ステージへ移行するときにその選択された命令が実行されて生成されるであろう結果データは確定的に保持されるべきという第1の処置情報を生成し、その対応するコミット条件中のいずれかの条件の真偽がすでに判定されている条件の真偽と対応的に一致しかつその対応するコミット条件中のその他の条件が未だ判定されていない第2の場合はその次の処理ステージへ移行するときにその選択された命令が実行されて生成されるであろう結果データは暫定的に保持されるべきという第2の処置情報を生成するステップと、その比較後、その選択された命令を実行して結果データを生成するステップと、上記第1の場合はその第1の処置情報に従ってその生成された結果データを確定的に保持し、上記第2の場合はその第2の処置情報に従ってその生成された結果データを暫定的に保持するステップと、上記第2の場合は、その対応するコミット条件中のすべての条件の真偽がすでに判定されている条件の真偽と対応的に一致するコミット時に、その暫定的に保持した結果データを確定的に保持し直すステップとを含む。

【0021】請求項5に記載のコミット条件付き命令の処理方法は、上記請求項4に記載の処理方法に加えてさらに、第1または第2の処置情報を生成した後、上記1つの処理ステージ以降のもう1つの処理ステージにおいて、その対応するコミット条件を上記条件と再び比較するステップと、その比較の結果、上記第2の場合から上記第1の場合へ変化している場合はさらにその次の処理ステージへ移行するときにその第2の処置情報を第1の

処置情報へ変更するステップを含む。

【0022】請求項6に記載のコミット条件付き命令の処理装置は、複数の条件の下で、各々が対応する1つのコミット条件を持つ複数の命令を処理するものを前提とする。ここで、上記コミット条件は、対応する命令を最終的に処理するために必要な上記条件のうちいずれかからなる。上記処理装置は、演算手段と、データ保持手段と、真偽レジスタ手段と、実行制御手段と、コミット制御手段とを備える。上記演算手段は、上記命令のうち選択された少なくとも1つの命令を実行し、かつ、上記条件のうち選択された少なくとも1つの条件を判定する。上記データ保持手段は、上記演算手段がその選択された命令を実行して生成した結果データを確定的または暫定的に保持する。上記真偽レジスタ手段は、上記複数の条件に対応する複数のエントリを含む。上記エントリの各々は、対応する条件が真であるという第1の条件情報、対応する条件が偽であるという第2の条件情報、および対応する条件の真偽が未定であるという第3の条件情報のうちいずれかを保持する。上記実行制御手段は、その対応するコミット条件を上記真偽レジスタ手段から得られた条件情報と比較し、(i) その対応するコミット条件中のすべての条件の真偽がそのすべての条件に対応する条件情報と対応的に一致する第1の場合は上記データ保持手段が結果データを確定的に保持するように制御し、(ii) その対応するコミット条件中のいずれかの条件の真偽がそのいずれかの条件に対応する条件情報と対応的に一致しかつその対応するコミット条件中のその他の条件に対応する条件情報が第3の条件情報である第2の場合は上記データ保持手段が結果データを暫定的に保持するように制御する。上記コミット制御手段は、その対応するコミット条件を上記真偽レジスタ手段から得られた条件情報と比較し、その対応するコミット条件中のすべての条件の真偽がそのすべての条件に対応する条件情報と対応的に一致するコミット時に、上記データ保持手段が暫定的に保持した結果データを確定的に保持し直すように制御する。

【0023】請求項7に記載のコミット条件付き命令の処理装置は、複数の条件の下で、各々が対応する1つのコミット条件を持つ複数の命令を複数の処理ステージを経てパイプライン制御によって処理するものを前提とする。ここで、上記コミット条件は、対応する命令を最終的に処理するために必要な上記条件のうちいずれかからなる。上記処理装置は、演算手段と、データ保持手段と、真偽レジスタ手段と、情報生成手段と、書込制御手段と、コミット制御手段とを備える。上記演算手段は、上記命令のうち選択された少なくとも1つの命令を実行し、かつ、上記条件のうち選択された少なくとも1つの条件を判定する。上記データ保持手段は、上記演算手段がその選択された命令を実行して生成した結果データを確定的または暫定的に保持する。上記真偽レジスタ手段

は、上記複数の条件に対応する複数のエントリを含む。上記エントリの各々は、対応する条件が真であるという第1の条件情報、対応する条件が偽であるという第2の条件情報、および対応する条件の真偽が未定であるという第3の条件情報のうちいずれかを保持する。上記情報生成手段は、上記複数の処理ステージのうち1つの処理ステージにおいて、その対応するコミット条件を上記真偽レジスタ手段から得られた条件情報と比較し、(i) その対応するコミット条件中のすべての条件の真偽がそのすべての条件に対応する条件情報と対応的に一致する第1の場合はその次の処理ステージへ移行するときに上記演算手段がその選択された命令を実行して生成するであろう結果データは確定的なものであるという第1の処置情報を生成し、(ii) その対応するコミット条件中のいずれかの条件の真偽がそのいずれかの条件に対応する条件情報と対応的に一致しかつその対応するコミット条件中のその他の条件に対応する条件情報が第3の条件情報である第2の場合はその次の処理ステージへ移行するときに上記演算手段がその選択された命令を実行して生成するであろう結果データは暫定的なものであるという第2の処置情報を生成する。上記書込制御手段は、

(i) 上記第1の場合はその第1の処置情報に従って上記演算手段が生成した結果データを上記データ保持手段が確定的に保持するように制御し、(ii) 上記第2の場合はその第2の処置情報に従って上記演算手段が生成した結果データを上記データ保持手段が暫定的に保持するように制御する。上記コミット制御手段は、その対応するコミット条件を上記真偽レジスタ手段から得られた条件情報と比較し、その対応するコミット条件中のすべての条件の真偽がそのすべての条件に対応する条件情報と対応的に一致するコミット時に、上記データ保持手段が暫定的に保持した結果データを確定的に保持し直すように制御する。

【0024】請求項8に記載のコミット条件付き命令の処理装置は、上記請求項7に記載の処理装置に加えてさらに、上記情報生成手段が第1または第2の処置情報を生成した上記1つの処理ステージ以降のもう1つの処理ステージにおいて、上記第2の場合から上記第1の場合へ変化している場合はさらにその次の処理ステージへ移行するときにその第2の処置情報を第1の処置情報へ変更する情報変更手段を備える。

【0025】請求項9に記載のコミット条件付き命令の処理装置は、上記請求項6～請求項8に記載の処理装置において、上記データ保持手段が、第1および第2のレジスタ手段を備えて構成されている。上記第1のレジスタ手段は、その結果データを確定的に保持する。上記第2のレジスタ手段と、その結果データを暫定的に保持する。

【0026】請求項10に記載のコミット条件付き命令の処理装置は、上記請求項6～請求項8に記載の処理装

置において、上記データ保持手段が、その結果データを確定的に保持しているのかまたは暫定的に保持しているのかを特定するためのフラグを備えて構成されている。

【0027】請求項11に記載のコミット条件付き命令の処理装置は、請求項6～請求項10に記載の処理装置において、上記条件のうち少なくとも1つの条件を判定するための条件命令もまた、対応する1つのコミット条件を持つ。上記真偽レジスタ手段の各エントリは、条件情報を確定的または暫定的に保持する。上記実行制御手段は、その条件命令に対応するコミット条件を上記真偽レジスタ手段から得られた条件情報と比較し、(i) その条件命令に対応するコミット条件中のすべての条件の真偽がそのすべての条件に対応する条件情報と対応的に一致する第3の場合は上記演算手段がその条件命令に従ってその選択された条件を判定して生成した第1または第2の条件情報を上記真偽レジスタ手段が確定的に保持するように制御し、(ii) その条件命令に対応するコミット条件中のいずれかの条件の真偽がそのいずれかの条件に対応する条件情報と対応的に一致しかつその条件命令に対応するコミット条件中のその他の条件に対応する条件情報が第3の条件情報である第4の場合は上記演算手段がその条件命令に従ってその選択された条件を判定して生成した第1または第2の条件情報を上記真偽レジスタ手段が暫定的に保持するように制御する。上記処理装置はさらに、その条件命令に対応するコミット条件を上記真偽レジスタ手段から得られた条件情報と比較し、その条件命令に対応するコミット条件中のすべての条件の真偽がそのすべての条件に対応する条件情報と一致するコミット時に、上記真偽レジスタ手段が暫定的に保持した第1または第2の条件情報を確定的に保持し直すように制御する第2のコミット制御手段を備える。

【0028】請求項12に記載のコミット条件付き命令の処理装置は、請求項6～請求項11に記載の処理装置に加えてさらに、例外記憶手段と、例外処理手段とを備える。例外記憶手段は、その選択された命令が処理される間に例外が発生しかつ上記第2の場合はその例外の発生を記憶する。例外処理手段は、上記コミット時に上記例外記憶手段がその例外の発生を記憶している場合はその例外を処理する。

【0029】請求項13に記載のコミット条件付き命令の処理装置は、上記例外処理手段がさらに、その例外を処理する前に、上記複数の命令のうち、少なくともその例外の発生時から上記コミット時までの間にその結果データが暫定的に保持されている命令を再処理する再処理手段を備えて構成されている。

【0030】請求項14に記載のコミット条件付き命令の処理装置は、上記請求項13に記載の処理装置において、上記再処理手段がその対応する結果データが暫定的に保持されている命令を逐次的な順序で再処理するように構成されている。

【0031】請求項15に記載のコミット条件付き命令の処理装置は、上記請求項6～請求項11に記載の処理装置に加えてさらに、真偽計算手段と、例外記憶手段と、将来真偽レジスタ手段と、再処理手段と、例外処理手段とを備える。上記真偽計算手段は、上記真偽レジスタ手段から得られた条件情報に上記演算手段がその選択された条件を判定して生成した条件情報を加える。上記例外記憶手段は、その選択された命令が処理される間に例外が発生しかつ上記第2の場合はその例外の発生を記憶する。上記将来真偽レジスタ手段は、上記真偽計算手段から得られた条件情報を保持する。上記例外処理手段は、上記コミット時に上記例外記憶手段がその例外の発生を記憶している場合は上記複数の命令のうち上記コミット時まですでに処理された命令を最初から再処理する。上記例外処理手段は、上記再処理手段がその命令を再処理している間にその例外が再び発生したとき、その例外を引起した命令に対応するコミット条件を上記将来真偽レジスタ手段から得られた条件情報と比較し、その対応するコミット条件中のすべての条件の真偽がそのすべての条件に対応する条件情報と対応的に一致する場合はその例外を処理する。上記真偽レジスタ手段は、上記再処理手段がその命令を再処理している場合は上記真偽計算手段から得られた条件情報を受入れず、そうでない場合はその条件情報を受入れる。上記真偽レジスタ手段はまた、上記再処理手段がその命令を再処理し終えたとき上記将来真偽レジスタ手段から得られた条件情報を受入れる。

【0032】請求項16に記載のコミット条件付き命令の処理装置は、上記請求項15に記載の処理装置において、上記再処理手段が、その命令のうち上記第2の場合における命令を選択的に再処理するように構成されている。

【0033】請求項17に記載のコミット条件付き命令の処理装置は、上記請求項12～請求項16に記載の処理装置において、上記例外記憶手段がその例外の発生を記憶している間はその記憶されている内容が書換えられるのを禁止する禁止手段をさらに備えて構成されている。

【0034】

【作用】請求項1に記載のコミット条件付き命令の処理方法においては、命令が実行される前に、コミット条件がすでに成立しているか否かが判断され、たとえそのコミット条件の成否が未だ判明していない場合であってもその命令は投機的に実行され、その結果データは暫定的に保持される。そして、そのコミット条件が成立したときその暫定的に保持された結果データは改めて確定的に保持される。このように、コミット条件の成否が判明する前に命令を実行できるので、命令の処理速度が向上する。

【0035】請求項2に記載のコミット条件付き命令の

処理方法においては、投機的に実行された命令が例外を引起した場合はコミット時にその例外処理が行なわれる。したがって、無駄な例外処理が行なわれることはなく、命令の処理速度はさらに向上する。

【0036】請求項3に記載のコミット条件付き命令の処理方法においては、コミット時まですでに処理された命令が再処理され、そして、再び例外が発生したときにその例外処理が行なわれる。したがって、投機的に実行された命令が例外を引起した場合であっても常に正確な結果データが生成される。

【0037】請求項4に記載のコミット条件付き命令の処理方法においては、命令はパイプライン制御によって複数の処理ステージを経て処理される。ここで、ある処理ステージにおいて、ある命令が逐次的かまたは投機的かという処置情報が生成される。これにより、そのコミット条件の成否が判明する前であってもその命令は投機的に実行され、その結果データは暫定的に保持される。暫定的に保持された結果データは、コミット時に改めて確定的に保持される。このように、命令は投機的に実行され得るので、命令の処理速度がさらに向上する。

【0038】請求項5に記載のコミット条件付き命令の処理方法においては、さらに別のある処理ステージにおいて、たとえばコミット条件の成立が判明した場合はその命令は逐次的であるという処置情報に変更され、その変更された処置情報に従ってその結果データは確定的に保持される。

【0039】請求項6に記載のコミット条件付き命令の処理装置においては、命令が実行される前に、コミット条件がすでに成立しているか否かが判断され、たとえコミット条件の成否が未だ判明していない場合であっても、その命令は投機的に実行され、その結果データは暫定的に保持される。そして、その暫定的に保持された結果データはコミット時に改めて確定的に保持される。このように、そのコミット条件の成否が未だ判明していない命令であっても投機的に実行され得るので、命令の処理速度は向上する。

【0040】請求項7に記載のコミット条件付き命令の処理装置においては、命令はパイプライン制御によって複数の処理ステージを経て処理される。ここで、ある処理ステージにおいて、その命令は逐次的であるかまたは投機的であるかという処置情報が生成される。その命令が投機的に実行されて生成された結果データは、その処置情報に従って暫定的に保持される。そして、その暫定的に保持された結果データはコミット時に改めて確定的に保持される。したがって、コミット条件の成否が未だ判明していない命令であっても投機的に実行され得るので、命令の処理速度はさらに向上する。

【0041】請求項8に記載のコミット条件付き命令の処理装置においては、さらに別のある処理ステージにおいて、たとえばその命令が確定的であることが判明した

場合はその処置情報を変更され、その変更された処置情報に従ってその結果データは確定的に保持される。

【0042】請求項9に記載のコミット条件付き命令の処理装置においては、命令が逐次的に実行された場合はその結果データは第1のレジスタ手段に確定的に保持され、一方、命令が投機的に実行された場合はその結果データは第2のレジスタ手段に暫定的に保持される。

【0043】請求項10に記載のコミット条件付き命令の処理装置においては、命令が逐次的に実行された場合はフラグが第1の状態でその結果データは確定的に保持される。一方、命令が投機的に実行された場合はフラグが第2の状態でその結果データは暫定的に保持される。

【0044】請求項11に記載のコミット条件付き命令の処理装置においては、条件命令が実行される前に、そのコミット条件がすでに成立しているか否かが判断され、そのコミット条件が未だ判明していない場合であっても、その条件命令は投機的に実行され、その条件情報が暫定的に保持される。したがって、通常の命令だけでなく、条件命令も投機的に実行され得る。

【0045】請求項12に記載のコミット条件付き命令の処理装置においては、命令が投機的に実行されたときに例外が発生した場合は、そのコミット時に例外処理が行なわれる。したがって、無駄な例外処理が行なわれることがなく、命令の処理速度はさらに向上する。

【0046】請求項13に記載のコミット条件付き命令の処理装置においては、コミット時までの命令が再処理され、再び例外が発生したときにその例外処理が行なわれる。したがって、常に正しい結果データが生成される。

【0047】請求項14に記載のコミット条件付き命令の処理装置においては、コミット時までの命令が逐次的な順序で処理されるので、命令の処理速度はさらに向上する。

【0048】請求項15に記載のコミット条件付き命令の処理装置においては、コミット時までの命令が最初から再処理される。そして、再び例外が発生したときに将来真偽レジスタ手段が参照され、その例外を引起した命令が将来逐次的なものになる場合はその例外処理が行なわれる。このように、コミット時までの命令が再処理され、再び例外が発生したとき必要に応じてその例外処理が行なわれるので、プログラムのバイナリサイズの増加が抑えられ、しかも正確な結果データが生成される。

【0049】請求項16に記載のコミット条件付き命令の処理装置においては、命令が再処理される場合は、逐次的な命令は再処理されずに投機的な命令だけが再処理されるので、例外処理は正確に行われる。

【0050】請求項17に記載のコミット条件付き命令の処理装置においては、例外の発生が一旦記憶された場合はその内容が書換えられることはないので、例外の発生はコミット時まで確実に記憶され続け、その例外処理

が確実に行なわれる。

【0051】

【実施例】次に、この発明に従った命令処理方法およびその装置の実施例を図面を参照して詳しく説明する。なお、図中同一符号で示される部分は同一または相当部分を示す。

【0052】【実施例1】図1は、この発明の第1実施例によるマイクロプロセッサの構成を示すブロック図である。

10 【0053】図1を参照して、このマイクロプロセッサは、4つのALU2Aないし2Dと、メモリ3と、実行すべき命令のアドレスをメモリ3に与えるプログラムカウンタ4と、プログラムカウンタ4をインクリメントする回路5と、メモリ3からフェッチされた命令を保持する命令レジスタ6と、命令レジスタ6に保持された命令を解釈し、ALU2Aないし2Dなどを制御するための制御信号を生成する命令デコーダ7と、分岐先アドレスを計算する回路9とを備える。

20 【0054】このマイクロプロセッサはさらに、図28に示した従来のマイクロプロセッサと異なり、真偽レジスタ(TFレジスタ)10と、実行制御回路11と、シーケンシャルレジスタファイル12と、シャドウレジスタファイル13と、コミット制御回路14とを備える。

30 【0055】図2は、このマイクロプロセッサによって処理され得る命令コードの構成を示す。図2を参照して、この命令コードは、命令操作部15およびコミット条件部16から構成される。命令操作部15は、従来のマイクロプロセッサによって処理され得る命令コードに相当する部分である。コミット条件部16は、命令操作部15に示された命令が実行されるために必要な1または2以上の条件を特定する部分である。

【0056】図3は、TFレジスタ10の構成を示す。図3を参照して、このTFレジスタ10は、m個のエントリを備える。各エントリは有効性ビットTvおよび値ビットTdの2ビットから構成され、ある1つの条件は真であるという第1の条件情報、その条件は偽であるという第2の条件情報、およびその条件の真偽は未定であるという第3の条件情報のうちいずれかを保持する。

40 【0057】表1を参照して、有効性ビットTvが「1」で、かつ値ビットTdが「1」の場合、その条件は真であることを意味する。また、有効性ビットTvが「1」で、値ビットTdが「0」の場合、その条件は偽であることを意味する。さらに、有効性ビットTvが「0」の場合、値ビットTdがいかなる値であってもその条件の真偽は未定であることを意味する。

【0058】

【表1】

有効性T _v	値T _d	意味
1	1	真
1	0	偽
0	X	未定

X: 任意の値

【0059】たとえば図2に示した命令コードは、TFレジスタ10における第1エントリが「真」で、かつ第2エントリも「真」の場合、レジスタr14が保持する値とレジスタr15が保持する値とを加算し、その結果データをレジスタr6に格納するという命令が本質的に実行されるべきであるということをマイクロプロセッサに指示することができる。

【0060】図4は、コミット条件部16の構成を示す。図4を参照して、このコミット条件部16は、TFレジスタ10におけるエントリに対応してm個のエントリを備える。各エントリは、TFレジスタ10と同様に、有効性ビットC_vおよび値ビットC_dの2ビットから構成される。

【0061】表2を参照して、有効性ビットC_vが「1」で、かつ値ビットC_dが「1」の場合は、対応するTFレジスタ10におけるエントリが「真」であることが、その命令操作部15に示された命令が本質的に実行されるための必要条件の1つであることを意味する。換言すれば、そのことは、その命令操作部15に示された命令が本質的に実行されるための必要条件のすべてではないことを意味する。

【0062】

【表2】

有効性C _v	値C _d	意味
1	1	真
1	0	偽
0	X	Don't Care

X: 任意の値

【0063】また、有効性ビットC_vが「1」で、かつ値ビットC_dが「0」の場合は、対応するTFレジスタ10におけるエントリが「偽」であるということが、その命令操作部15に示された命令が本質的に実行されるための必要条件の1つであることを意味する。

【0064】さらに、有効性ビットC_vが「0」の場合は、値ビットC_dに関係なく、しかも対応するTFレジスタ10におけるエントリが「真」、「偽」および「未定」のいずれであっても、その命令操作部15に示された命令は常に実行されるべきものであるということを意味する。

【0065】実行制御回路11は、コミット制御部16のエントリとTFレジスタ10のエントリとを対応的に比較し、その結果に応じてALU2Aないし2Dが確定的に命令を実行するように制御するか、または暫定的に

命令を実行するように制御する。

【0066】図5は、この実行制御回路11の機能をさらに詳しく説明するための概念図である。

【0067】図5を参照して、この命令Iは3つのある条件C1ないしC3の下で実行されるべきものである。たとえば、第1の条件C1が「真」で、第2の条件C2が「真」で、かつ第3の条件C3が「偽」の場合だけ命令Iは本質的に実行されるとすると、コミット条件部16における第1のエントリは「真」を保持し、第2のエントリは「真」を保持し、第3のエントリは「偽」を保持している。

【0068】これらの条件C1ないしC3に対応するTFレジスタ10における第1ないし第3のエントリが「真」または「偽」をそれぞれ保持していて、かつその真偽と、第1ないし第3の条件C1ないしC3の真偽とが対応的に一致している場合(A)、実行制御回路11はALU2Aないし2Dが命令Iを確定的に実行するように制御する。この場合における命令Iを特に「逐次的命令」という。

【0069】また、第1ないし第3の条件C1ないしC3に対応するTFレジスタ10における第1ないし第3のエントリが「未定」をそれぞれ保持している場合

(B)、実行制御回路11はALU2Aないし2Dが暫定的に命令Iを実行するように制御する。この場合における命令Iを特に「投機的命令」という。

【0070】また、第1ないし第3の条件C1ないしC3に対応するTFレジスタ10における第1ないし第3のエントリのうちいくつかは「真」または「偽」をそれぞれ保持していて、かつその真偽と第1ないし第3の条件C1ないしC3の真偽とが対応的に一致している場合(C)、実行制御回路11はALU2Aないし2Dが命令Iを暫定的に実行するように制御する。この場合における命令Iも「投機的命令」という。

【0071】シーケンシャルレジスタファイル12およびシャドウレジスタファイル13は、それぞれ32個のレジスタ30および31から構成される。シーケンシャルレジスタファイル12は、従来のマイクロプロセッサにおけるレジスタファイル1に相当するものである。

【0072】ただし、このシーケンシャルレジスタファイル12は、ALU2Aないし2Dが命令を実行して生成した結果データを確定的に保持する。シャドウレジスタファイル13は、ALU2Aないし2Dが命令を実行して生成した結果データを暫定的に保持する。

【0073】コミット制御回路14は、シャドウレジスタファイル13に格納された結果データに対応する命令のコミット条件部16を保持する。コミット制御回路14はさらに、TFレジスタ10を常に監視していて、コミット条件部16におけるエントリとTFレジスタ10におけるエントリとを比較し、それらが対応的に一致している場合、シャドウレジスタファイル13に格納され

た結果データをシーケンシャルレジスタファイル12へ転送する。

【0074】たとえば図5を参照して、投機的命令Iが実行され、その結果データが暫定的にシャドウレジスタファイル13に格納された場合、命令Iのコミット条件部16はコミット制御回路14の中に格納される。

【0075】そして、命令Iが実行されるために必要な第1ないし第3の条件C1ないしC3に対応するTFレジスタ10における第1ないし第3のエントリが「真」または「偽」を保持した時点で、その真偽と命令Iが実行されるために必要な第1ないし第3の条件C1ないしC3の真偽とが対応的に一致している場合、コミット制御回路14はシャドウレジスタファイル13に格納された結果データをシーケンシャルレジスタファイル12へ転送する。

【0076】ここで、実行制御回路11、シーケンシャルレジスタファイル12、シャドウレジスタファイル13およびコミット制御回路14の構成をさらに詳細に説明する。

【0077】図6は、実行制御回路11の構成を示すブロック図である。図6を参照して、実行制御回路11は、コミット条件計算回路17と、インパータ18と、*

*ANDゲート19Aと、ORゲート19Bとを備える。

【0078】図7は、コミット条件計算回路17の構成を示すブロック図である。図7を参照して、コミット条件計算回路17は、比較回路20および21と、ANDゲート22および23とを備える。なお、図7では、コミット条件部16およびTFレジスタ10がそれぞれ2つのエントリを備える場合に対応する、コミット条件計算回路が示されている。

【0079】図8は、コミット条件計算回路17における比較回路20または21の構成を示す回路図である。図8を参照して、比較回路20または21は、インパータ24と、NANDゲート25と、排他的NORゲート26と、NANDゲート27および28とを備える。

【0080】比較回路20または21は、コミット条件部16における1つのエントリとそれに対応するTFレジスタ10における1つのエントリとを比較し、その結果に応じて有効性ビットV_iおよび値ビットD_iを生成する。

【0081】表3は、比較回路20または21の動作を示す真理値表である。

【0082】

【表3】

コミット条件部における第iエントリ		TFレジスタにおける第iエントリ		第iエントリの計算結果	
Cv	Cd	Tv	Td	有効性Vi	値Di
0	X	X	X	1	1
1	X	0	X	0	X
1	Cd	1	Td	1	Cd=Td

X: 任意の値

【0083】表3を参照して、コミット条件部における1つのエントリの有効性ビットCvが「0」の場合、それに対応するTFレジスタ10における1つのエントリの有効性ビットTvおよび値ビットTdがいかなる値であっても、比較回路20または21は「1」の有効性ビットV_iと、「1」の値ビットD_iとを生成する。

【0084】また、コミット条件部16における1つのエントリの有効性ビットCvが「1」であり、かつそれに対応するTFレジスタ10における1つのエントリの有効性ビットTvが「0」の場合、それらエントリの値ビットCdおよびTdがいかなる値であっても、比較回路20または21は「0」の有効性ビットV_iを生成する。

【0085】さらに、コミット条件部16における1つのエントリの有効性ビットCvと、それに対応するTFレジスタ10における1つのエントリの有効性ビットTvとがそれぞれ「1」であり、それらの値ビットCdおよびTdが互いに等しい場合、比較回路20または21は「1」の有効性ビットV_iと、「1」の値ビットD_iとを生成する。それらエントリの値ビットCdおよびT

dが異なる場合、比較回路20または21は、「1」の有効性ビットV_iと、「0」の値ビットD_iとを生成する。

【0086】すべての比較回路20および21の有効性ビットV_iが「1」の場合だけ、コミット条件計算回路17は「1」の有効性ビットVを生成する。また、すべての比較回路20および21の値ビットD_iが「1」の場合だけ、コミット条件計算回路17は「1」の値ビットDを生成する。

【0087】したがって、コミット条件計算回路17の有効性ビットVが「0」の場合、「1」のシャドウレジスタ書込指定ビットSDが生成される。また、コミット条件計算回路17の有効性ビットVおよび値ビットDがともに「1」の場合か、または有効性ビットVが「0」の場合、「1」の命令有効性ビットIVが生成される。

【0088】以上詳述したところから明らかなように、コミット条件部16におけるすべてのエントリが「Don't Care」を保持している場合（それに対応する命令操作部における命令は無条件下で実行されるべきものである場合）、実行制御回路11は「0」のシャドウレジ

タ書込指定ビットSDを生成するとともに、「1」の命令有効性ビットIVを生成する。

【0089】また、コミット条件部16におけるいくつかのエントリが「真」または「偽」を保持していて、それらに対応するTFレジスタにおけるエントリがそれぞれ「未定」を保持している場合、実行制御回路11は「1」のシャドウレジスタ書込指定ビットSDを生成するとともに、「1」の命令有効性ビットIVを生成する。

【0090】さらに、コミット条件部16におけるいくつかのエントリが「真」または「偽」を保持していて、かつそれらに対応するTFレジスタ10におけるエントリのうちいくつかが「真」または「偽」を保持していて、かつそれらコミット条件部16における真偽とTFレジスタ10における真偽とが対応的に一致している場合は、実行制御回路11は「0」のシャドウレジスタ書込指定ビットSDを生成するとともに、「1」の命令有効性ビットIVを生成する。

【0091】命令有効性ビットIVが「1」の場合、生成された結果データはシーケンシャルレジスタファイル12に格納される。また、シャドウレジスタ書込指定ビットSDが「0」であり、かつ命令有効性ビットIVが「0」の場合、命令は実行されないか、または命令が実行されることによって生成された結果データはシーケンシャルレジスタファイル12およびシャドウレジスタ13のいずれにも格納されない。さらに、シャドウレジスタ書込指定ビットSDが「1」で、かつ命令有効性ビットIVが「1」の場合、命令が実行されることによって生成された結果データは暫定的にシャドウレジスタファイル13に格納される。

【0092】図9は、シーケンシャルレジスタファイル12、シャドウレジスタファイル13およびコミット制御回路14の構成を示すブロック図である。

【0093】図9を参照して、シーケンシャルレジスタファイル12は、32個のシーケンシャルレジスタ30から構成される。シャドウレジスタファイル13は、32個のシャドウレジスタ31から構成される。

【0094】コミット制御回路14は、32個のシーケンシャルレジスタ30およびシャドウレジスタ31に対応して32個のエントリから構成される。各エントリは、コミット条件記憶部32と、コミット条件計算回路17と、ANDゲート33とを備える。

【0095】コミット条件記憶部32は、投機的命令が実行され、その結果データがシャドウレジスタファイル13におけるシャドウレジスタ31に暫定的に格納された場合、その命令に対応するコミット条件部16を保持する。

【0096】コミット条件計算回路17は、前述した実行制御回路11におけるコミット条件計算回路17と同一で、コミット条件記憶部32に保持されたコミット条

件部16とTFレジスタ10とを比較し、その結果に応じて有効性ビットVと値ビットDとを生成する。ANDゲート33はコミット条件計算回路17の有効性ビットVと値ビットDとの論理積をシーケンシャルレジスタ30とシャドウレジスタ31との間に接続されたトランスファergeート34のゲートに与える。

【0097】コミット条件計算回路17はTFレジスタ10を常に監視していて、コミット条件記憶部32が保持するコミット条件部16における真偽と、TFレジスタ10における真偽とが対応的に一致したとき、「1」の有効性ビットVと、「1」の値ビットDとを生成する。これによりトランスファergeート34はオンになり、シャドウレジスタ31における結果データがシーケンシャルレジスタ30に転送される。

【0098】次に、この第1実施例によるマイクロプロセッサの動作を図10に示したプログラムに従って説明する。

【0099】図11は、このマイクロプロセッサを図10に示したプログラムに従って動作させるためのプログラムリストである。

【0100】図11において、各行は複数の命令コードから構成される。また、各命令コードは命令操作部15およびコミット条件部16から構成される。同一行を構成する命令コードの命令操作部15に示される命令は同時に実行される。

【0101】なお、命令コードは互いに「;」によって分離されている。また、命令操作部15およびコミット条件部16は「?」によって分離されている。

【0102】コミット条件部16に示される「always」はTFレジスタ10における真偽に関係なく、常にその命令操作部15に示される命令I1、I2、C1またはC2が無条件で実行されるべきものであることを示す。

【0103】また、TF[i]は、TFレジスタ10における第iエントリが「真」であるときだけその命令操作部15に示された命令が実行されるべきものであることを示す。また、/TF[i]は、TFレジスタ10における第iエントリが「偽」であるときだけその命令操作部15に示される命令が実行されるべきものであることを示す。

【0104】さらに、たとえばTF[1]&T[2]は、TFレジスタ10における第1および第2エントリがともに「真」であるときだけその命令操作部に示される命令が実行されるべきものであることを示す。また、TF[1]&/TF[2]は、TFレジスタ10における第1エントリが「真」であり、かつ第2エントリが「偽」であるときだけその命令操作部15に示される命令が実行されるべきものであることを示す。

【0105】図11におけるブロックK1、L1~L3は、図10における基本ブロックK1、L1ないしL3に対応する。そして、このブロックK1、L1~L3か

ら3つの基本ブロックM1ないしM3が分岐している。

【0106】まず第1サイクルにおいて、第1行目の4つの命令I1ないしI4が同時に実行される。

【0107】命令コード「always? I1」によれば、命令I1は無条件で実行される。すなわち、この命令コードのコミット条件部16におけるすべてのエントリは「don't care」であり、各エントリの有効性ビットCvは「0」である。

【0108】したがって、コミット条件計算回路17は、TFレジスタ10における各エントリの有効性ビットTvおよびTdに関係なく、ともに「1」の有効性ビットVおよび値ビットDを生成する。これにより実行制御回路11は「0」のシャドウレジスタ書込指定ビットSDと「1」の命令有効性ビットIVとを生成する。

【0109】そのため、シーケンシャルレジスタファイル12におけるレジスタr10およびr11のデータが加算され、その結果データはシーケンシャルレジスタファイル12におけるレジスタr2に確定的に格納される。

【0110】同時に命令コード「TF[1]? I2」によれば、この命令コードのコミット条件部16における第1エントリは「真」であり、その有効性ビットCvおよび値ビットCdはともに「1」である。また、コミット条件部16における第1エントリ以外の有効性ビットCvはすべて「0」である。

【0111】この時点で、TFレジスタ10における第1エントリは「未定」であり、その有効性ビットTvは「0」であるため、実行制御回路11は「1」のシャドウレジスタ書込指定ビットSDを生成し、かつ「0」の命令有効性ビットIVを生成する。したがって、シーケンシャルレジスタファイル12におけるレジスタr12およびr13のデータは加算され、その結果データはシャドウレジスタファイル13におけるレジスタr4に暫定的に格納される。

【0112】このとき、命令レジスタ6に保持されているコミット条件部16のデータは、その結果データが格納されたシャドウレジスタr4に対応するコミット制御回路14における第4エントリのコミット条件記憶部32に格納される。

【0113】同時に命令コード「TF[1]? I3」によれば、この命令コードのコミット条件部16における第1エントリは「偽」であり、その有効性ビットCvは「1」である。前述したようにTFレジスタ10における第1のエントリは「未定」であるので、命令I3に従ってシーケンシャルレジスタファイル12におけるレジスタr12のデータからレジスタr13のデータが減算され、その結果データがシャドウレジスタファイル13におけるシャドウレジスタr5に暫定的に格納される。

【0114】このときもまた、コミット条件部16のデ

ータはその結果データが格納されたシャドウレジスタr5に対応するコミット制御回路14における第5エントリのコミット条件記憶部32に格納される。

【0115】同時に命令コード「TF[1]&TF[2]? I4」によれば、この命令コードのコミット条件部16の第1および第2エントリはともに「真」である。このとき、前述したようにTFレジスタ10における第1および第2エントリは「未定」であるので、命令I4に従ってシーケンシャルレジスタファイル12におけるレジスタr14およびr15のデータは加算され、その結果データはシャドウレジスタファイル13におけるレジスタr6に暫定的に格納される。

【0116】このときもまた、コミット条件部16のデータはコミット制御回路14における第6エントリのコミット条件記憶部32に格納される。

【0117】次いで第2サイクルにおいて、命令コード「always? C1」に従って条件命令C1が常に行実行される。すなわち、もしレジスタr2のデータがレジスタr3のデータよりも小さければTFレジスタ10における第1エントリが「真」にされ、そうでなければ「偽」にされる。

【0118】また同時に、命令コード「always? C2」に従って、レジスタr4およびr5のデータが互いに等しければTFレジスタ10における第2エントリが「真」にされ、そうでなければ「偽」にされる。

【0119】このとき、コミット制御回路14はTFレジスタ10を監視し、コミット条件記憶部32に格納されたコミット条件部16とTFレジスタ10とを比較している。

【0120】もしTFレジスタ10における第1エントリが「真」であれば、コミット制御回路14における第4エントリのコミット条件計算回路17はともに「1」の有効性ビットVおよび値ビットDを生成する。これら有効性ビットVおよび値ビットDの論理積がANDゲート33によってトランスファゲート34のゲートに与えられ、このトランスファゲート34はオンになる。これによりシャドウレジスタファイル13におけるレジスタr4のデータがトランスファゲート34を介して対応するシーケンシャルレジスタファイル12におけるレジスタr4へ転送される。

【0121】このとき、シャドウレジスタファイル13におけるレジスタr5のデータは何らの処理もされない。一方、もしTFレジスタ10における第1エントリが「偽」であれば、シャドウレジスタファイル13におけるレジスタr5のデータがトランスファゲート34を介して対応するシーケンシャルレジスタファイル12におけるレジスタr5へ転送される。

【0122】このとき、シャドウレジスタファイル13におけるレジスタr4のデータは何らの処理もされな

い。

【0123】また、もしTFレジスタ10における第1および第2エントリがともに「真」であれば、シャドウレジスタファイル13におけるレジスタr6のデータがトランスファークラウド34を介して対応するシーケンシャルレジスタファイル12におけるレジスタr6へ転送される。

【0124】しかしながら、もしTFレジスタ10における第1および第2エントリのうち少なくとも一方が「偽」であれば、シャドウレジスタファイル13におけるレジスタr6のデータは何らの処理もされない。

【0125】次いで第3サイクルにおいて、命令コード「TF[1]&/TF[2]?J1」に従って、もしTFレジスタ10における第1エントリが「真」であり、かつ第2エントリが「偽」であれば、無条件分岐命令J1が実行される。すなわち、もし条件C1が「真」であり、かつ条件C2が「偽」であれば基本ブロックM2を指定するアドレスがプログラムカウンタ4に格納される。

【0126】もしTFレジスタ10における第1エントリが「偽」であれば、無条件分岐命令J2が実行される。すなわち、もし条件C1が「偽」であれば、基本ブロックM1を指定するアドレスがプログラムカウンタ4に格納される。

【0127】もしTFレジスタ10における第1および第2エントリがともに「真」であれば、無条件分岐命令J3が実行される。すなわち、もし条件C1およびC2がともに「真」であれば、基本ブロックM3を指定するアドレスがプログラムカウンタ4に格納される。

【0128】以上、この第1実施例によれば、本質的に1または2以上の条件下で実行されるべき命令が、その条件の真偽が判明する前にその命令が実行される。たとえば図11に示したプログラムによれば、命令I2は条件C1の下で実行されるべきにもかかわらず、その条件C1の真偽が判明する前の第1サイクルにおいて投機的に実行される。また、命令I3も条件C1の下で実行されるべきにもかかわらず、その条件C1の真偽が判明する前の第1サイクルにおいて投機的に実行される。さらに、命令I4は、条件C1およびC2の下で実行されるべきにもかかわらず、それら条件C1およびC2の真偽が判明する前の第1サイクルにおいて投機的に実行される。

【0129】したがって、従来のマイクロプロセッサであれば第1サイクルにおいて実行することができない投機的命令I2ないしI4が実行され、ALU2Aないし2Dが有効に活用される。このため、命令の処理速度は向上する。

【0130】なお、この第1実施例において、シーケンシャルレジスタファイル12およびシャドウレジスタファイル13は、結果データを確定的または暫定的に保持するデータ保持手段に相当する。

【0131】〔実施例2〕図12は、この発明の第2実施例によるマイクロプロセッサの構成を示すブロック図である。この第2実施例によるマイクロプロセッサは、上記第1実施例によるマイクロプロセッサがパイプライン化されたものである。

【0132】図12を参照して、このマイクロプロセッサは、本質的に1または2以上の条件下で実行されるべき複数の命令を4つの処理ステージS1ないしS4を経て処理する。

10 【0133】このマイクロプロセッサは、上記第1実施例によるマイクロプロセッサと同様に、4つのALU2Aないし2Dと、メモリ3と、プログラムカウンタ4と、インクリメント回路5と、命令レジスタ6と、命令デコーダ7と、分岐先アドレス計算回路9と、TFレジスタ10と、実行制御回路11と、シーケンシャルレジスタファイル12と、シャドウレジスタファイル13と、コミット制御回路14とを備える。

20 【0134】図13は、このマイクロプロセッサによる動作を示すタイムチャートである。図12および図13を参照して、第1処理ステージS1では、プログラムカウンタ4によって指定された命令がメモリ3からフェッチされ、命令レジスタ6に格納される。

【0135】第2処理ステージS2では、命令レジスタ6に格納された命令が命令デコーダ7によって解釈され、かつ命令デコーダ7からの制御信号に応答してシーケンシャルレジスタファイル12またはシャドウレジスタファイル13から所定のデータが読出される。

30 【0136】このマイクロプロセッサはさらに、命令デコーダ7によって生成されたALU2Aないし2Dを制御するための信号を格納するためのレジスタ35と、シーケンシャルレジスタファイル12またはシャドウレジスタファイル13から読出されたデータを格納するためのレジスタ36および37とを備える。

【0137】第3処理ステージS3では、ALU制御信号レジスタ35に格納された制御信号に従って、レジスタ36および37に格納されたデータがALU2Aないし2Dによって演算される。

40 【0138】第4処理ステージS4では、ALU2Aないし2Dによって生成された結果データがシーケンシャルレジスタファイル12またはシャドウレジスタファイル13に書込まれ、またALU2Aないし2Dによって生成された真または偽の条件情報がTFレジスタ10に書込まれ、さらにコミット条件部16におけるデータがコミット制御回路14に書込まれる。

【0139】次に、コミット条件付きの命令をパイプライン制御に従って処理する、このマイクロプロセッサ特有の構成について説明する。

50 【0140】図12を参照して、このマイクロプロセッサはさらに、実行制御回路11と、コミット条件レジスタ38と、第1の有効性レジスタ39と、第1の書込指

定レジスタ40とを備える。

【0141】実行制御回路11は上記第1実施例におけるものと同一で、コミット条件部16とTFレジスタ10とを比較し、その結果に従って命令有効性ビットIVとシャドウレジスタ書込指定ビットSDとを生成する。

【0142】コミット条件レジスタ38は、ALU2Aないし2Dにおいて実行中の命令に対応するコミット条件部16を保持する。第1の有効性レジスタ39は、実行制御回路11によって生成された、その命令に対応する命令有効性ビットIVを保持する。第1の書込指定レジスタ40は、実行制御回路11によって生成された、その命令に対応するシャドウレジスタ書込指定ビットSDを保持する。

【0143】このマイクロプロセッサはさらに、コミット条件計算回路41と、有効性計算回路42と、書込指定計算回路43と、第2の有効性レジスタ44と、第2の書込指定レジスタ45とを備える。

【0144】コミット条件計算回路41は図7に示した*

コミット条件計算結果			命令有効性ビットの処置
V	D	意味	
1	1	コミット条件は満足されている	そのままの値で次の処理ステージへ
1	0	コミット条件は満足されていない	「0」にする
0	X	コミット条件が満足されているか否か不明	そのままの値で次の処理ステージへ

【0148】表4を参照して、コミット条件計算回路41がともに「1」の有効性ビットVおよび値ビットDを生成した場合、ORゲート48の出力は「1」になるので、第1の有効性レジスタ39に格納された命令有効性ビットIVはANDゲート49を介して第2の有効性レジスタ44へ転送される。

【0149】また、コミット条件計算回路41が「1」の有効性ビットVと「0」の値ビットDとを生成した場合、ORゲート48の出力は「0」になるので、「0」の命令有効性ビットIVが第2の有効性レジスタ44に格納される。

【0150】さらに、コミット条件計算回路41が「0」の有効性ビットVを生成した場合、ORゲート48の出力は「1」になるので、第1の有効性レジスタ3

*コミット条件計算回路17と同一で、コミット条件レジスタ38に格納されたコミット条件部16とTFレジスタ10とを比較し、その結果に従って有効性ビットVと値ビットDとを生成する。

【0145】図14は、有効性計算回路42の構成を示すブロック図である。図14を参照して、この有効性計算回路42は、コミット条件計算回路41からの有効性ビットVおよび値ビットDを受けるANDゲート46と、ANDゲート46の出力を受けるとともに、その有効性ビットVをインバータ47を介して受けるORゲート48と、ORゲート48の出力と第1の有効性レジスタ39からの命令有効性ビットIVとを受けるANDゲート49とを備える。

【0146】次の表4は、有効性計算回路42の機能を示す。

【0147】

【表4】

9に格納された命令有効性ビットIVはANDゲート49を介して第2の有効性レジスタ44へ転送される。

【0151】図15は、書込指定計算回路43の構成を示すブロック図である。図15を参照して、この書込指定計算回路43は、コミット条件計算回路41からの有効性ビットVを受けるインバータ50と、インバータ50の出力と第1の書込指定レジスタ40からのシャドウレジスタ書込指定ビットSDとを受けるANDゲート51とを備える。

【0152】次の表5は、この書込指定計算回路43の機能を示す。

【0153】

【表5】

コミット条件計算結果		シャドウレジスタ 書込指定ビットの処理
V	意 味	
1	コミット条件が満足 されているか否かは判明	「0」にする
0	コミット条件が満足 されているか否かは不明	そのままの値で次の処理ステージへ

【0154】表5を参照して、コミット条件計算回路41が「1」の有効性ビットVを生成した場合、インバータ50の出力は「0」になるので、「0」のシャドウレジスタ書込指定ビットSDが第2の書込指定レジスタ45に格納される。

【0155】また、コミット条件計算回路41が「0」の有効性ビットVを生成した場合、インバータ50の出力は「1」になるので、第1の書込指定レジスタ40に格納されたシャドウレジスタ書込指定ビットSDはANDゲート51を介して第2の書込指定レジスタ45へ転送される。

【0156】このマイクロプロセッサはさらに、レジスタファイル書込制御回路52を備える。このレジスタファイル書込制御回路52は、第2の有効性レジスタ44に格納された命令有効性ビットIVが「1」で、かつ第2の書込指定レジスタ45に格納されたシャドウレジスタ書込指定ビットSDが「0」の場合、その命令に従ってALU2Aないし2Dが生成した結果データはシーケンシャルレジスタファイル12に書込まれるように制御する。

【0157】また、第2の有効性レジスタ44に格納された命令有効性ビットIVが「1」で、かつ第2の書込指定レジスタ45に格納されたシャドウレジスタ書込指定ビットSDが「1」の場合、このレジスタファイル書込前記回路52は、その命令に従ってALU2Aないし2Dが生成した結果データはシャドウレジスタファイル13に書込まれるように制御する。

【0158】さらに、第2の有効性レジスタ44に格納された命令有効性ビットIVが「0」の場合、このレジスタファイル書込制御回路52は、その命令に従ってALU2Aないし2Dが生成した結果データはシーケンシャルレジスタファイル12およびシャドウレジスタファイル13のいずれにも書込まれないように制御する。

【0159】次に、この第2実施例によるマイクロプロセッサの動作について図12および図13を参照して説明する。

【0160】まず第1クロックで、第1処理ステージS1におけるメモリ3から最初の命令がフェッチされ、命令レジスタ6に格納される。

【0161】次いで第2クロックで、第2処理ステージS2における命令デコーダ7によって命令レジスタ6に格納された最初の命令が解読されるとともに、その命令

デコーダ7からの制御信号に従ってシーケンシャルレジスタファイル12またはシャドウレジスタファイル13におけるデータが読出され、レジスタ36または37に格納される。また、第2処理ステージS2における実行制御回路11によってその第1の命令に対応する命令有効性ビットIVおよびシャドウレジスタ書込指定ビットSDが生成され、第1の有効性レジスタ39および第1の書込指定レジスタ40にそれぞれ格納される。

【0162】すなわち、コミット条件部16におけるエントリの真偽とTFレジスタ10におけるエントリの真偽とが対应的に一致している場合、この実行制御回路11は、この第1の命令に従って後に生成されることになる結果データはシーケンシャルレジスタファイル12に格納されるべきものであるという処置情報を第3処理ステージS3に移行するときに生成する。

【0163】また、コミット条件部16におけるエントリに対応するTFレジスタ10におけるエントリのうち少なくとも1つが「未定」で、それ以外のエントリのうちコミット条件部16におけるエントリに対応するTFレジスタ10におけるエントリが「真」または「偽」で、かつその真偽とコミット条件部16におけるエントリの真偽とが対应的に一致している場合、実行制御回路11は、この第1の命令に従って後に生成されることになる結果データはシャドウレジスタファイル13に格納されるべきものであるという処置情報を第3の処理ステージS3に移行するときに生成する。

【0164】さらに、コミット条件部16におけるエントリの真偽とTFレジスタ10におけるエントリの真偽とが対应的に一致していない場合、実行制御回路11は、この第1の命令に従って後に生成されることになる結果データはシーケンシャルレジスタファイル12およびシャドウレジスタファイル13のいずれにも格納されるべきものではないという処置情報を第3の処理ステージS3に移行するときに生成する。

【0165】これと同時に、第1の処理ステージS1ではメモリ3から第2の命令がフェッチされ、命令レジスタ6に格納される。

【0166】次いで第3クロックで、ALU制御信号レジスタ35に格納された制御信号に従って、第3処理ステージS3におけるALU2Aないし2Dは第1の命令を実行する。また、第3処理ステージS3におけるコミット条件計算回路41は、コミット条件レジスタ38に

格納された第1の命令に対応するコミット条件部16とTFレジスタ10とを比較し、第1の命令のための有効性ビットVおよび値ビットDを生成する。

【0167】第3処理ステージS3における有効性計算回路42はコミット条件計算回路41からの有効性ビットVおよび値ビットDに従って、第1の有効性レジスタ39に格納された命令有効性ビットIVを第2の有効性レジスタ44へそのまま転送するか、またはその命令有効性ビットIVを変更する。

【0168】また、書込指定計算回路43は、コミット条件計算回路41からの有効性ビットVに従って、第1の書込指定レジスタ40に格納されたシャドウレジスタ書込指定ビットSDを第2の書込指定レジスタ45へそのまま転送するか、またはそのシャドウレジスタ書込指定ビットSDを変更する。

【0169】たとえば第1の有効性レジスタ39に「1」の命令有効性ビットIVが格納され、かつ第1の書込指定レジスタ40に「0」のシャドウレジスタ書込指定ビットSDが格納されている場合（第1の命令は有効で、かつその結果データはシーケンシャルレジスタファイル12に書込まれるべきということが既に判明している場合）、コミット条件計算回路41はともに「1」の有効性ビットVおよび値ビットDを常に生成するため、「1」の命令有効性ビットIVはそのまま第2の有効性レジスタ44に格納されるとともに、「0」のシャドウレジスタ書込指定ビットSDは「0」にされる。結果的に、シャドウレジスタ書込指定ビットSDはそのまま第2の書込指定レジスタ45に格納される。

【0170】また、「0」の命令有効性ビットIVが第1の有効性レジスタ39に格納され、かつ「0」のシャドウレジスタ書込指定ビットSDが第1の書込指定レジスタ40に格納されている場合（第1の命令は既に無効と判明している場合）、コミット条件計算回路41は「1」の有効性ビットVと「0」の値ビットDとを常に生成するため、「0」の命令有効性ビットIVおよび「0」のシャドウレジスタ書込指定ビットSDはともに「0」にされる。結果的に、命令有効性ビットIVおよびシャドウレジスタ書込指定ビットSDはともに、そのまま第2の有効性レジスタ44および第2の書込指定レジスタ45にそれぞれ格納される。

【0171】また、「1」の命令有効性ビットIVが第1の有効性レジスタ39に格納され、かつ「1」のシャドウレジスタ書込指定ビットSDが第1の書込指定レジスタ40に格納されている場合（第1の命令に従って生成されることになる結果データはシャドウレジスタファイル13に格納されるべき場合）は、第3処理ステージS3において、その第1の命令に対応するコミット条件は満たされていることが判明する場合、そのコミット条件は満たされていないことが判明する場合、およびそのコミット条件が満たされているか否かは未だ判明しない

場合のいずれかが起こり得る。

【0172】コミット条件は満たされていることが判明した場合（コミット条件計算回路41がともに「1」の有効性ビットVおよび値ビットDを生成した場合）、

「1」の命令有効性ビットIVはそのまま第2の有効性レジスタ44に格納され、「1」のシャドウレジスタ書込指定ビットSDは「0」に変更され、第2の書込指定レジスタ45に格納される。

【0173】また、コミット条件は満たされていないことが判明した場合（コミット条件計算回路41が「1」の有効性ビットVと「0」の値ビットDとを生成した場合）、ともに「1」の命令有効性ビットIVおよびシャドウレジスタ書込指定ビットSDはともに「0」に変更され、第2の有効性レジスタ44および第2の書込指定レジスタ45にそれぞれ格納される。

【0174】さらに、コミット条件が満たされているか否かは未だ判明していない場合（コミット条件計算回路41が「0」の有効性ビットVを生成した場合）、ともに「1」の命令有効性ビットIVおよびシャドウレジスタ書込指定ビットSDはそのまま第2の有効性レジスタ44および第2の書込指定レジスタ45にそれぞれ格納される。

【0175】またこれと同時に、命令レジスタ6にフェッチされた第2の命令は命令デコーダ7によって解読され、その命令デコーダ7からの制御信号に従ってシーケンシャルレジスタファイル12またはシャドウレジスタファイル13におけるデータが読出される。

【0176】さらにこれと同時に、第3の命令がメモリ3からフェッチされる。次いで第4クロックで、第1の命令に従ってALU2Aないし2Dが生成した結果データは、レジスタファイル書込制御回路52からの制御信号に従って、シーケンシャルレジスタファイル12およびシャドウレジスタファイル13のいずれかに書込まれる。すなわち、命令有効性ビットIVが「1」で、かつシャドウレジスタ書込指定ビットSDが「0」の場合、レジスタファイル書込制御回路52は、結果データがシーケンシャルレジスタファイル12に書込まれるように制御する。

【0177】また、命令有効性ビットIVが「1」で、かつシャドウレジスタ書込指定ビットSDが「1」の場合、レジスタファイル書込制御回路52は、その結果データがシャドウレジスタファイル13に書込まれるように制御する。

【0178】さらに、命令有効性ビットIVが「0」の場合、レジスタファイル書込制御回路52は、その結果データがシーケンシャルレジスタファイル12およびシャドウレジスタファイル13のいずれにも書込まれないように制御する。

【0179】なお、結果データがシャドウレジスタファイル13に書込まれる場合は、その第1の命令に対応す

るコミット条件部16がコミット制御回路14におけるコミット条件記憶部32に格納される。

【0180】またこれと同時に、第2の命令がALU2Aないし2Dによって実行され、その命令有効性ビットIVおよびシャドウレジスタ書込指定ビットSDが第2の有効性レジスタ44および第2の書込指定レジスタ45にそれぞれ格納される。

【0181】またこれと同時に、第3の命令が命令デコーダ7によって解読され、その命令デコーダ7からの制御信号に従ってシーケンシャルレジスタファイル12またはシャドウレジスタファイル13におけるデータが読出される。

【0182】さらにこれと同時に、第4の命令がメモリ3からフェッチされる。以下、第2ないし第4の命令は第1の命令と同様に処理される。

【0183】なお、この第2実施例において、第2処理ステージS2における実行制御回路11は情報生成手段に相当する。また、コミット条件計算回路41、有効性計算回路42および書込指定計算回路43は、情報変更手段に相当する。さらに、レジスタファイル書込制御回路52は書込制御手段に相当する。

【0184】〔実施例3〕上記第1実施例によるマイクロプロセッサは、たとえば図10に示したプログラムに従うと、条件命令C1を実行する前に、条件命令C2を実行することができる。しかしながら、このプログラムの意味上は、条件命令C1が実行された後に、条件命令C2は実行されるべきものである。

【0185】このプログラムの意味にしたがって、条件命令C1を実行した後でなければ条件命令C2を実行することができないように構成されたマイクロプロセッサであってもよい。

【0186】図16は、この第3実施例によるマイクロプロセッサのためのプログラムの一例を示すプログラムリストである。なお、命令I1ないしI4、C1およびC2、J1ないしJ3は、図11に示されるものと同一である。

【0187】ここで特徴的なところは、条件命令C2が条件命令C1の実行結果に依存して実行されている点である。すなわち、条件命令C2はTFレジスタ10における第1エントリが「真」であるときだけ実行される。このTFレジスタ10における第1エントリの真偽は、その1つの前のサイクルにおいて条件命令C1が実行されることによって決定される。

【0188】したがって、条件命令C1が実行され、TFレジスタ10における第1エントリに「真」が書込まれた場合だけ条件命令C2が実行され、これによりTFレジスタ10における第2エントリに「真」または「偽」が書込まれる。

【0189】換言すれば、上記第1実施例における第2エントリは単に条件命令C2の実行結果を含むだけであ

ったのに対し、この第3実施例における第2エントリは、条件命令C2の実行結果だけでなく、第1エントリが「真」であるということを暗に含んでいる。

【0190】このため、上記第1実施例における「TF[0]&TF[2]」という条件は単に「TF[2]」と表わされる。

【0191】図17は、第3実施例によるマイクロプロセッサの実行制御回路およびコミット制御回路におけるコミット条件計算回路の構成を示すブロック図である。

10 【0192】図17を参照して、このコミット条件計算回路53は、比較回路20と、デコーダ54と、2つのセレクト55および56とを備える。このコミット条件計算回路53はコミット制御部58とTFレジスタ10とを比較し、その結果に応じて有効性ビットVおよび値ビットDを生成する。

【0193】コミット条件部58は、有効性ビットと、値ビットと、比較されるべきTFレジスタ10のエントリ番号を示すビットとから構成される。

20 【0194】TFレジスタ10は上記第1実施例におけるものと同一である。ただし、図16に示したプログラムの場合、最大で2つの条件を超えて命令が実行されるので、TFレジスタ10は2つのエントリを持っていれば十分である。

【0195】一方、コミット条件計算回路53における比較回路20は上記第1実施例におけるものと同一である。また、デコーダ54はTFレジスタ10のエントリ番号を解読し、所定の選択信号を生成する。セレクト55は、デコーダ54からの選択信号に応答してTFレジスタ10における第1または第2エントリの有効性ビットを選択し、その選択された有効性ビットTvを比較回路20に与える。セレクト56は、デコーダ54からの選択信号に応答してTFレジスタ10における第1または第2エントリの値ビットを選択し、その選択された値ビットTdを比較回路20に与える。

30 【0196】たとえば命令コード「TF[1]?I2」によれば、命令I2はTFレジスタ10における第1エントリが「真」の場合だけ実行されるべきものである。この場合、コミット条件部58のTFレジスタ10のエントリ番号を示すビットは第1エントリを特定する。これにより、TFレジスタ10における第1エントリの有効性ビットTvおよび値ビットTdがセレクト55および56によってそれぞれ選択され、比較回路20に与えられる。

40 【0197】また、命令コード「TF[2]?I4」によれば、コミット条件部58におけるTFレジスタ10のエントリ番号を示すビットはTFレジスタ10における第2エントリを特定する。これにより、TFレジスタ10における第2エントリの有効性ビットTvおよび値ビットTdがセレクト55および56によってそれぞれ選択され、比較回路20に与えられる。

【0198】この第3実施例によれば、コミット条件部58、コミット条件計算回路53が上記第1実施例のものと比べて簡略化される。

【0199】なお、この第3実施例によるマイクロプロセッサを上記第2実施例のようにパイプライン化してもよい。

【0200】〔実施例4〕図18は、この発明の第4実施例によるマイクロプロセッサの構成を示すブロック図である。

【0201】図18を参照して、このマイクロプロセッサは、上記第1実施例と異なり、TFレジスタ10の代わりに、シーケンシャルTFレジスタ60と、シャドウTFレジスタ61と、TFレジスタ用コミット制御回路62とを備える。

【0202】シーケンシャルTFレジスタ60は上記第1実施例におけるTFレジスタ10に相当し、ある条件が「真」であるか、「偽」であるか、またはその真偽は「未定」であるかの3つの情報のうちいずれかを確定的に保持する1または2以上のエントリを備える。

【0203】シャドウTFレジスタ61は、ある条件が「真」であるか、「偽」であるか、またはその真偽は「未定」であるかの3つの情報のうちいずれかを暫定的に保持する1または2以上のエントリを備える。

【0204】TFレジスタ用のコミット制御回路62は、「未定」のコミット条件を持つ条件命令が実行された場合、そのコミット条件を保持するとともに、そのコミット条件の真偽と、それに対応するシーケンシャルTFレジスタ10の真偽とが一致した時点で、シャドウTFレジスタ61に格納された「真」または「偽」をシーケンシャルTFレジスタ60へ転送する。

【0205】なお、このTFレジスタ用のコミット制御回路62は、上記第1実施例におけるコミット制御回路14とほぼ同じ構成である。

【0206】図19は、このマイクロプロセッサのためのプログラムの一例を示すプログラムリストである。

【0207】このプログラムに従うと、第2サイクルにおいて命令コード「TF[1]?C2」が処理される。この命令コード「TF[1]?C2」によれば、条件命令C2はシーケンシャルTFレジスタ60における第1エントリが「真」である場合だけ実行されるべきものである。

【0208】しかしながら、この第1エントリの真偽は次の第3サイクルにおける命令コード「always?C1」が処理されたとき初めて決定される。

【0209】したがって、この第2サイクルの時点ではシーケンシャルTFレジスタ60における第1エントリは「未定」であるので、条件命令C2は暫定的に実行され、その真偽はシャドウTFレジスタ61における第2エントリに格納される。これと同時に、コミット条件「TF[1]」はTFレジスタ用のコミット制御回路6

2におけるコミット条件記憶部に格納される。

【0210】次いで第3サイクルにおける命令コード「always?C1」によれば、条件命令C1は常に行われ、その真偽はシーケンシャルTFレジスタ60における第1エントリに格納される。

【0211】TFレジスタ用のコミット制御回路62は常にシーケンシャルTFレジスタ60を監視している。そのため、このTFレジスタ用のコミット制御回路62におけるコミット条件記憶部に保持されたコミット条件の真偽と、シーケンシャルTFレジスタ60における第1エントリの真偽とが一致している場合は、シャドウTFレジスタ61における第2エントリに格納されている条件命令C2の真偽がシーケンシャルTFレジスタ60における第2エントリに転送される。

【0212】この第4実施例によれば、コミット条件が「未定」の単純な演算命令I2ないしI3だけでなく、コミット条件が「未定」の条件命令C2をも投機的に実行することができる。

【0213】なお、この第4実施例において、シーケンシャルTFレジスタ60およびシャドウTFレジスタ61は、真偽レジスタ手段に相当する。また、コミット制御回路14は、第1のコミット制御手段に相当する。また、TFレジスタ用コミット制御回路62は第2のコミット制御手段に相当する。

【0214】〔実施例5〕図20は、この発明の第5実施例によるマイクロプロセッサにおけるレジスタファイルの構成を示す図である。

【0215】図20を参照して、このマイクロプロセッサは、32個のレジスタRA1ないしRA32から構成される第1のレジスタファイル65と、32個のレジスタRB1ないしRB32から構成される第2のレジスタファイル66と、32個のフラグF1ないしF32とを備える。

【0216】この第5実施例によれば、コミット条件が「未定」であるため、その命令が投機的に実行されたときの結果データは第1のレジスタファイル65における1つのレジスタRA1～RA32に暫定的に格納され、かつそれに対応するフラグF1～F32が「0」にされる。

【0217】そして、そのコミット条件の真偽が判明し、その真偽とTFレジスタにおける真偽とが対応的に一致した時点で、そのフラグは「1」にされる。

【0218】上記第1実施例のようにシャドウレジスタファイル13からシーケンシャルレジスタファイル12へ物理的にデータを転送しなくても、この第5実施例のようにフラグを反転することによってレジスタファイル65および66を投機的状態から逐次的状態に変更するようにしてもよい。

【0219】〔実施例6〕上記第1ないし第5実施例によるマイクロプロセッサは、投機的な命令を実行するこ

とができるが、投機的な命令が実行される時点ではその結果データが有効かどうかは不明であるため、投機的な命令が実行されたときに例外が発生した場合、その例外を処理すべきかどうかという問題が生じる。

【0220】最も簡単な方法として、投機的な命令が例外を引き起こしたときも、逐次的な命令が例外を引き起こしたときと同様に、その例外を引き起こした時点でその例外を処理するという方法が考えられる。

【0221】この方法は非常に簡単ではあるが、命令の処理効率を著しく低下させるという問題がある。たとえば前述したページフォールトという例外が引き起こされた場合、それに対応する例外処理を行なうには数十万サイクルが必要とされる。しかしながら、このページフォールトを引き起こした投機的な命令が実行されて生成された結果データが不必要な場合、この数十万サイクルは全く無駄になる。

【0222】図21は、このような問題を解決するためになされた、この発明の第6実施例によるマイクロプロセッサの構成を示すブロック図である。

【0223】図21を参照して、このマイクロプロセッサは、上記第1実施例によるマイクロプロセッサと同様に、4つのALU2Aないし2Dと、メモリ3と、プログラムカウンタ4と、インクリメント回路5と、命令レジスタ6と、命令デコーダ7と、分岐先アドレス計算回路9と、TFレジスタ10と、実行制御回路11と、シーケンシャルレジスタファイル12と、シャドウレジスタファイル13と、コミット制御回路14とを備える。

【0224】このマイクロプロセッサはさらに、上記第1実施例によるマイクロプロセッサと異なり、例外プログラムカウンタ(EPC; Excepted Program Counter)70と、セクタ71と、EP(Excepted Pending)回路72とを備える。

【0225】EPC70は、メモリ3からフェッチされた命令が例外を引き起こしたとき、そのプログラムカウンタ4に格納されているアドレスを記憶する。セクタ71は、ALU2Aないし2Dによって生成された結果データか、またはEPC70に格納されたアドレスを選択してシーケンシャルレジスタファイル12か、またはシャドウレジスタファイル13へ供給する。なお、EPC70およびセクタ71は従来のマイクロプロセッサにも備えられている。

【0226】EP回路72は、投機的な命令が例外を引き起こしたことを示す例外発生信号SEにตอบสนองして、その投機的な命令が例外を引き起こしたことを記憶するとともに、所定の時点でその例外を処理すべきであることを示す例外処理信号RSEを生成する。

【0227】この例外発生信号SEは、その例外の種類に応じて様々な箇所で生成される。たとえば例外がページフォールトであれば、例外発生信号SEはメモリ3の周辺回路によって生成される。また、例外がオーバフロ

一であれば、例外発生信号SEはALU2Aないし2Dによって生成される。また、定義されていない命令がフェッチされた場合であれば、例外発生信号SEは命令デコーダ7によって生成される。

【0228】図22は、シーケンシャルレジスタファイル12、シャドウレジスタファイル13、コミット制御回路14およびEP回路72の構成を示すブロック図である。

【0229】シーケンシャルレジスタファイル12、シャドウレジスタファイル13およびコミット制御回路14は上記第1実施例におけるものとほぼ同一構成である。

【0230】EP回路72は、シーケンシャルレジスタファイル12、シャドウレジスタファイル13およびコミット制御回路14と同様に、32個のエントリから構成される。このEP回路72における各エントリは、ラッチ回路74と、ANDゲート75と、NチャネルMOSトランジスタ76とを備える。

【0231】このラッチ回路74は、入力端子IN、イネーブル端子Eおよびリセット端子Rを備え、その入力端子INへバッファ77を介して与えられた例外発生信号SEにตอบสนองして「1」または「0」を格納する。ラッチ回路74の出力信号はそれ自身のイネーブル端子Eへ与えられる。したがって、Hレベルの例外発生信号SEがバッファ77を介してラッチ回路74の入力端子INへ与えられると、このラッチ回路74に「1」が設定される。このとき、Hレベルの出力信号が負論理のイネーブル端子Eへ与えられるので、新たに例外発生信号SEが与えられてもこのラッチ回路74が書換えられることはない。

【0232】ラッチ回路74の出力信号およびコミット制御回路14におけるANDゲート33の出力信号はANDゲート75の入力端子へそれぞれ与えられ、このANDゲート75の出力信号はNチャネルMOSトランジスタ76のゲートへ与えられる。

【0233】ここで、32個のエントリにおける32個のNチャネルMOSトランジスタ76、プルアップ抵抗79およびインバータ80によりORゲート81が構成される。

【0234】また、コミット条件記憶部32もまた入力端子INおよびイネーブル端子Eを備え、投機的命令が実行されたとき、バッファ78を介してその入力端子INへ与えられたそのコミット条件部16を記憶する。ラッチ回路74の出力信号は、コミット条件記憶部32のイネーブル信号Eへも与えられている。したがって、例外が発生し、ラッチ回路74に「1」が格納されると、コミット条件記憶部32も書換えられることができない。

【0235】次に、この第6実施例によるマイクロプロセッサの動作について説明する。図23は、以上詳述し

た第6実施例によるマイクロプロセッサの動作の概要を示すタイミングチャートである。

【0236】逐次の命令および投機的命令をともに含む主プログラムに従って様々な命令が順次処理される。このとき、投機的命令が実行されて例外が発生したとしてもその例外処理はコミット点まで行なわれない。

【0237】また、各コミット点に対応して、ソフトウェアによって作成されたリカバリコードが予め用意されている。そして、このコミット点に到達すると、このリカバリコードの最初の命令から実行される。

【0238】このリカバリコードは、主プログラムの中からコミット点以前の投機的命令だけが抽出されて構成されている。しかもそれらの投機的命令は逐次的に配列され、逐次の命令に変更されている。そして、以前に例外を引き起こした投機的命令が実行されると、以前と同様に再び例外が発生する。しかしながら、今度は直ちに所定の例外ルーチンに従ってその例外処理が実行される。そしてリカバリコードにおける最後の命令が実行されると、主プログラムに戻りそのコミット点の次の命令から処理が再開される。

【0239】次に、この第6実施例によるマイクロプロセッサの動作をより詳細に説明する。

【0240】たとえば図11に示されるプログラムが実行される場合において、コミット条件付きの命令TF

【1】? I2が例外を引き起こすと仮定する。

【0241】まず第1サイクルにおいて、逐次の命令I1が実行され、レジスタr10に格納されている値とレジスタr11に格納されている値とが加算され、その結果データがシーケンシャルレジスタファイル12における第2エントリのシーケンシャルレジスタr2に格納される。

【0242】またこれと同時に、投機的命令I2が実行される。この命令I2のコミット条件TF【1】に対応するTFレジスタ10における第1エントリは「未定」であるので、この命令I2は投機的に実行される。これにより、レジスタr12に格納されている値とレジスタr13に格納されている値とが加算され、その結果データがシャドウレジスタファイル13における第4エントリのシャドウレジスタr4に格納される。

【0243】この命令I2は例外を引き起こすので、Hレベルの例外発生信号SEがバッファ77を介してEP回路72における第4エントリのラッチ回路74に与えられる。これにより、第4エントリのラッチ回路74には「1」が書込まれる。なお、この命令I2は例外を引き起こしているため、その結果データは任意の値になっている。

【0244】またこれと同時に、コミット条件付きの命令I3およびI4が実行される。これらの命令I3およびI4はともに投機的であるので、その結果データはシャドウレジスタファイル13における第5および第6エ

ントリのシャドウレジスタ31に格納される。

【0245】しかしながら、これらの命令I3およびI4は例外を引き起こさないで、Lレベルの例外発生信号SEがEP回路72における第5および第6エントリのラッチ回路74に与えられる。これにより、これらのラッチ回路74には「0」が格納される。

【0246】次いで第2サイクルにおいては、逐次の条件命令C1およびC2が同時に実行される。ここで、もし第2エントリのシーケンシャルレジスタr2に格納されている値が第3エントリのシーケンシャルレジスタr3に格納されている値よりも小さければ、TFレジスタ10における第1エントリに「真」が設定される。これにより、コミット制御回路14における第4エントリのコミット条件記憶部32に格納されている内容とTFレジスタ10に格納されている内容とが一致すると、そのANDゲート33の出力信号がHレベルになる。

【0247】そのため、第4エントリにおけるトランスファゲート34がオンになり、第4エントリのシャドウレジスタ31に格納されている命令I2の結果データが第4エントリのシーケンシャルレジスタ30へ転送される。これと同時に、EP回路72における第4エントリのANDゲート75が導通状態になり、そのラッチ回路74からの出力信号がNチャネルMOSトランジスタ76のゲートへ与えられる。

【0248】第1サイクルにおいて命令I2が実行されたとき例外が発生しているため、EP回路72における第4エントリのラッチ回路74には「1」が設定されている。したがって、ラッチ回路74の出力信号はHレベルになり、NチャネルMOSトランジスタ76はオンになる。そのため、例外処理信号RSEはHレベルになる。このHレベルの例外実行信号RSEにตอบสนองして、例外ハンドラの開始アドレスがプログラムカウンタ4に設定される。これにより例外処理が実行される。

【0249】一方、もし第2エントリのシーケンシャルレジスタr2に格納されている値が第3エントリのシーケンシャルレジスタr3に格納されている値よりも小さくなければ、TFレジスタ10における第1エントリに「偽」が設定される。

【0250】この場合、コミット制御回路14における第4エントリのANDゲート33の出力信号はHレベルにならず、Lレベルのまま維持される。そのため、シャドウレジスタファイル13における第4エントリのシャドウレジスタ31に格納されている結果データは対応するシーケンシャルレジスタ30へ転送されない。また、ラッチ回路74からのHレベルの出力信号はNチャネルMOSトランジスタ76のゲートへ与えられない。そのため、例外処理信号RSEはHレベルにならず、Lレベルのまま維持される。したがって、例外処理は実行されない。

【0251】以上のように、このマイクロプロセッサ

は、投機的な命令 I 2 が実行されたとき例外が発生しても直ちにその例外処理は実行されず、その投機的な命令 I 2 のコミット条件が成立するコミット点まで例外処理が延期され、そのコミット点で初めて例外処理が実行される。

【0252】すなわち、このマイクロプロセッサは、コミット点で初めて以前に実行された投機的命令 I 2 が例外を引き起こしたことを知り、例外ハンドラを起動する。

【0253】ところで、投機的命令が実行されたとき例外が発生した場合は、無意味な結果データが生成される。プログラムによってはこのような無意味な結果データに基づいて他の投機的命令が実行されているかもしれないので、例外処理を実行した後、その例外を引き起こした投機的命令を再び実行するだけでは十分ではなく、それらの他の投機的命令も再び実行する必要がある。

【0254】また、本質的には例外が発生したときからコミット点までの投機的命令を再び実行すればよいのであるが、そのような投機的命令だけを主プログラムから抽出することは煩雑である。そこで、この第6実施例によるマイクロプロセッサは、コミット点以前のすべての投機的命令を再び実行するように構成されている。

【0255】なお、これらの再び実行されるべき投機的命令だけで構成されるルーチン（リカバリコード）は、コミット点ごとにプログラムとして予め設けられている。また、既にコミット条件の真偽は決定されているので、これら投機的命令を逐次的に実行することができるように、それらの順序は変更されている。

【0256】たとえば図11に示されたプログラムの場合、第2サイクルのコミット点においては、図24に示すようなリカバリコードがソフトウェアによって作成される。図24に示すように、逐次的命令 always? I 1 はこのリカバリコードに含まれていない。また、このリカバリコードの中のすべての命令は逐次的になるようにその順番が変更されている。

【0257】これを詳細に説明すると、条件命令 C 1 が実行されると、TFレジスタ10における第1エントリの真偽が決定されるので、この真偽をコミット条件部に持つ投機的命令 I 2 および I 3 は、条件命令 C 1 が実行された後に実行されるように配列され、逐次的命令にされている。さらに、条件命令 C 2 が実行されると、TFレジスタ10における第2エントリの真偽が決定されるので、これら第1および第2エントリの真偽をコミット条件部に持つ投機的命令 I 4 は、条件命令 C 1 および C 2 が実行された後に実行されるように配列され、逐次的命令にされている。

【0258】このリカバリコードに従って、命令 I 2 が再び実行されると、再び例外が発生する。ただし、この場合における命令 I 2 はTFレジスタ10における第1エントリの真偽が決定された後に実行される逐次的な命

令であるから、この例外処理は直ちに実行される。

【0259】また、リカバリコードの中のすべての命令は逐次的命令にされているので、投機的な例外が発生することはない。また、リカバリコードの最後には、コミット点の次のアドレスに復帰するためのジャンプ命令 jmp が設けられている。

【0260】次に、コミット点からリカバリコードを呼出す方法としては、たとえばコミット点のアドレスをインデックスとしてリカバリコードの開始アドレスを参照することができるジャンプテーブルをプログラムの中に用意する方法がある。この方法の場合、割込ハンドラの中に図25に示されるプログラムを実行するコードを格納しておけばよい。

【0261】このプログラムによれば、第1のステートメントにおいて特定のコミット点のアドレスに対応するリカバリコードの開始アドレスがジャンプテーブルから呼出され、変数「recoveryCodeAddress」に格納される。「jumpTable」は各コミット点のアドレスに対応するリカバリコードの開始アドレスが配列されたテーブルである。

【0262】次いで第2のステートメントにおいて、プログラムカウンタ4にそのリカバリコードの開始アドレスが格納され、そのリカバリコードに従って順次所定の命令が処理される。

【0263】一般に、EPCは割込を引き起こした命令のアドレスを保持するレジスタであるが、この第6実施例におけるEPC70はコミット点のアドレスを保持するレジスタである。すなわち、このEPC70は、例外処理信号RSEがHレベルになって割込みが生じたとき、そのアドレスを保持する。

【0264】最後に、EP回路72におけるラッチ回路74に「1」が設定されると、そのラッチ回路74自身およびそれに対応するコミット条件記憶部32の書換えが禁止されるように構成した理由について説明する。

【0265】もし書換えを禁止する手段がなければ、図26に示すようなプログラムを実行する場合に問題が生じる。

【0266】命令 I 10 が実行され、例外が発生したと仮定する。命令 I 10 は投機的であるので、直ちに例外処理は行なわれず、EP回路72における第1エントリのラッチ回路74に「1」が設定されるとともに、コミット制御回路14における第1エントリのコミット条件記憶部32にTF「1」というコミット条件が書込まれる。

【0267】このとき、シャドウレジスタファイル13における第1エントリのシャドウレジスタ31にその投機的命令が実行されて生成された結果データが書込まれるが、その命令は例外を引き起こしているため、その結果データは正しい値ではない。

【0268】次いで投機的命令 I 11 が実行されたとき

例外が発生しなかった場合、もしラッチ回路74に書込え禁止の手段がなければ、EP回路72における第1エントリのラッチ回路74に「0」が設定され、これにより命令I10が例外を引き起こしたという記憶が消去されてしまう。

【0269】したがって、EP回路72におけるラッチ回路74に「1」が設定された場合、その内容が他の命令によって「0」が再設定されることがないようにする必要はある。

【0270】このことはコミット条件記憶部32についても同様である。なぜならば、コミット条件記憶部32は投機的命令が例外を引き起こした場合にその例外処理を行なう条件を記憶しているのであるから、他の命令によって書換えられてはならないからである。

【0271】また、図26に示したプログラムの場合、命令I11は、命令I10の結果データr1に基づいて実行されている。この命令I10は例外を引き起こしている、その結果データr1は正しい値ではない。

【0272】したがって、例外を引き起こした投機的命令を再実行するだけでは十分でない。そこで、この第6実施例によるマイクロプロセッサは、コミット点よりも前の投機的命令をすべて再実行するように構成されている。

【0273】以上のようにこの第6実施例によれば、投機的命令が実行されたときに例外が発生しても直ちにその例外処理は実行されず、コミット点において必要なときだけ例外処理が実行されるので、無駄な例外処理が実行されることはない。そのため、より効率的にかつ高速に命令を処理することができる。

【0274】なお、この第6実施例において、ラッチ回路74は例外記憶手段に相当する。EPC70およびセ*

プログラムのバイナリサイズの増加率

$$\begin{aligned}
 &= \frac{\text{主プログラムのバイナリサイズ} + \text{ジャンプテーブルのサイズ} + \text{リカバリコードのサイズ}}{\text{主プログラムのバイナリサイズ}} \\
 &= \frac{(5\text{バイト} \times 4\text{命令}) + (4\text{バイト}) + (5\text{バイト} \times 4\text{命令})}{5\text{バイト} \times 4\text{命令}} \\
 &= \frac{20 + 4 + 20}{20} \\
 &= 2.2
 \end{aligned}$$

【0280】上記数1から明らかなように、ジャンプテーブルおよびリカバリコードを含む全プログラムのバイナリサイズは、主プログラムのみのバイナリサイズの約2.2倍になる。

【0281】また、上記第6実施例においては、例外が発生する確率は非常に低いにもかかわらず、ジャンプテーブルの参照とリカバリコードの実行によって余分に2

*レクタ71などは例外処理手段に相当する。

【0275】【実施例7】上記第6実施例によれば、無駄な例外処理が行なわれないので効率的に命令が処理されるが、各コミット点に設けられたリカバリコードと、そのリカバリコードを呼出すためのジャンプテーブルとが必要のため、プログラムのバイナリサイズが大きくなり、それによりディスクの記憶領域が大幅に占有されるという欠点がある。

【0276】ジャンプテーブルの各エントリにはリカバリコードの開始アドレスが書込まれているので、たとえば32ビットのマイクロプロセッサであれば、ジャンプテーブルの各エントリサイズは4バイトになる。ここで、並列に発行される命令の集合を命令ブロックと呼ぶこととする。リカバリコードの開始アドレスは各コミット点に対応して必要なので、ジャンプテーブルのサイズ上最悪になるのはすべての命令ブロックがコミット点となる場合である。

【0277】一方、リカバリコードには主プログラム中の投機的命令が逐次的に配列されているため、リカバリコードのサイズは主プログラムのサイズと同じかそれよりも若干大きくなる。ここで、大まかにリカバリコードのサイズを主プログラムのサイズと同じと仮定すると、たとえば5バイトの命令を4つ発行するマイクロプロセッサ、すなわち1つの命令ブロックが20バイトのマイクロプロセッサであれば、その1つの命令ブロックに対応するリカバリコードのサイズは20バイトになる。

【0278】したがって、主プログラムだけのバイナリサイズに対する全プログラムのバイナリサイズの増加率は、次の数1で表される。

【0279】
【数1】

度のキャッシュミスあるいはページフォールトが発生する確率が高く、それにより性能が低下する可能性が高いという問題もある。

【0282】さらに、リカバリコードでは命令が逐次的に並べ変えられているので、投機的な実行が可能なハードウェアが有効に活用されていないという問題もある。

【0283】図27は、上記のような問題を解消するた

めになされた、この発明の第7実施例によるマイクロプロセッサの全体構成を示すブロック図である。

【0284】図27を参照して、このマイクロプロセッサは、分岐命令の分岐先アドレスを格納するリカバリプログラムカウンタ(RPC)85と、各条件は「真」か「偽」かまたは「未定」かという条件情報を保持するTFレジスタ86と、TFレジスタ86から得られた条件情報にALU2A~2Dから得られた条件情報を加えて出力する真偽計算回路(TF計算回路)87と、TF計算回路87から得られた条件情報を保持する将来真偽レジスタ(FTFレジスタ)88と、例外処理信号RSE、割込信号などに応答して割込制御信号を生成する割込制御回路89と、プログラムカウンタ4の値をEPC70の値と比較し、それらが一致したときHレベルの比較信号CMPを生成する比較回路90と、Hレベルの比較信号CMPに応答してリセットされるリカバリモードビット92を持つステータスレジスタ91とを備える。

【0285】ここで、TFレジスタ86は、上述した実施例中のTFレジスタ10と異なり、例外処理信号RSEがLレベルの間だけTF計算回路87からの条件情報を受入れる。すなわち、例外処理信号RSEがHレベルの間はTFレジスタ10内の条件情報の書換えが禁止される。TFレジスタ86はまた、Hレベルの比較信号CMPに応答してFTFレジスタ88からの条件情報を受入れる。

【0286】TF計算回路87は、TFレジスタ86のエントリに対応してm個のエントリを含む。TF計算回路87の各エントリは、ALU2A~2Dから対応する条件情報が与えられない場合はTFレジスタ86の対応するエントリの条件情報をそのまま出力し、ALU2A~2Dから対応する条件情報が与えられた場合はその与えられた条件情報をそのまま出力する。

【0287】したがって、たとえばTFレジスタ86の第1および第2エントリが「未定」を保持しかつ第3エントリが「真」を保持しているときに、ALU2A~2Dから第1エントリに対応する条件は「真」であるという条件情報が供給された場合は、TF計算回路87は、その第1エントリから「真」を出力し、その第2エントリから「未定」を出力し、さらにその第3エントリから「真」を出力する。また、たとえばTFレジスタ86の第1エントリが「偽」を保持し、第2エントリが「真」を保持し、さらに第3エントリが「未定」を保持しているときに、ALU2A~2Dからその第3エントリに対応する条件は「偽」であるという条件情報が供給された場合は、TF計算回路87は、その第1エントリから「偽」を出力し、その第2エントリから「真」を出力し、さらにその第3エントリから「偽」を出力する。

【0288】FTFレジスタ88はTFレジスタ86と同様に、m個のエントリを含む。FTFレジスタ88の各エントリは、対応する条件が将来「真」になるのか、

将来「偽」になるのか、あるいはその条件の真偽は「未定」であるかのいずれかを保持する。

【0289】割込制御回路89は、従来のマイクロプロセッサにおける割込制御回路と同様に、割込信号に応答して割込制御信号を生成する。割込信号は、オーバフロー、ページフォールトなどの例外が発生したとき、その例外の発生部分から供給される。EP回路72から割込制御回路89に供給される例外処理信号RSEもまた、割込信号の1つである。したがって、割込制御回路89はまた、この例外処理信号RSEにも応答して割込制御信号を生成する。

【0290】割込制御信号が生成されると例外ハンドラが呼出され、その発生した例外に対応する例外処理が行なわれる。例外処理信号RSEに応答して割込制御信号が生成され、それにより例外ハンドラが呼出された場合は、ステータスレジスタ91のリカバリモードビット92が「1」にセットされる。これにより、後述するリカバリモードとなる。また、例外処理信号RSEに応答して呼出された例外ハンドラは、RPC85の値をプログラムカウンタ4に格納する。

【0291】割込制御回路89はまた、リカバリモードビット92が「1」にセットされている間に割込信号が発生したときは、処理中の命令が持っているコミット条件をFTFレジスタ88の条件情報と比較し、そのコミット条件が成立している場合は割込制御信号を生成し、それに応答して呼出された例外ハンドラは対応する例外処理を行なう。

【0292】次に、この第7実施例によるマイクロプロセッサの動作について説明する。上記第6実施例と同様に、図11に示されるプログラムが実行される場合において、コミット条件付きの命令I2が例外を引き起こすと仮定する。

【0293】ある分岐命令に従ってこのプログラム中の第1サイクルのアドレスがプログラムカウンタ4に格納されると、この基本ブロックK1へ実行が移行される。このときプログラムカウンタ4に格納されるアドレスは、RPC85にも格納される。

【0294】まず第1サイクルにおいて、4つの命令I1~I4が同時に実行される。ここで、命令I2が投機的に実行されると、レジスタr12に格納されている値とレジスタr13に格納されている値とが加算され、その結果データがシャドウレジスタファイル13における第4エントリのシャドウレジスタr4に格納される。

【0295】この命令I2は例外を引き起こすが、投機的であるので例外処理は直ちには行なわれない。その代わりに、EP回路72における第4エントリのラッチ回路74に「1」がセットされる。また、コミット制御回路14における第4エントリのコミット条件記憶部32にはそのコミット条件TF[1]が格納される。

【0296】続いて第2サイクルにおいては、2つの条

件命令C1、C2が逐次的に実行される。ここで、レジスタr2の値がレジスタr3の値よりも小さい場合は、ALU2A~2Dからの条件情報に従ってTF計算回路87の第1エン트리から「真」という条件情報が出力される。これにより、コミット制御回路14における第4エントリのコミット条件記憶部32に格納されているコミット条件TF[1]がTF計算回路87から与えられる条件情報と一致するため、EP回路72からの例外処理信号RSEはHレベルに立上がる。このHレベルの例外処理信号RSEに应答して割込制御回路89は、そのときのプログラムカウンタ4の値をEPC70に書込む。したがって、EPC70には、例外を引起こす命令I2のアドレスがそのまま書込まれるのではなく、その命令I2のコミット条件TF[1]を確定するための条件命令C2のアドレスが書込まれる。

【0297】TF計算回路87からの条件情報はFTFレジスタ88に書込まれる。レジスタr2の値がレジスタr3の値よりも小さく、しかもレジスタr4の値がレジスタr5の値と等しい場合は、TF計算回路87の第1および第2のエン트리からともに「真」という条件情報が出力される。したがって、FTFレジスタ88の第1および第2エントリにはともに「真」という条件情報が書込まれる。

【0298】このとき、例外処理信号RSEはHレベルであるため、TF計算回路87からのこれらの条件情報はTFレジスタ86には書込まれない。すなわち、上記実施例6ではこの時点でTFレジスタ10の第1および第2エントリが「真」となるが、この実施例7ではTFレジスタ86の第1および第2エントリは「未定」のまま維持され、その代わりにFTFレジスタ88の第1および第2エントリが「真」となる。

【0299】また、割込制御回路89からの割込制御信号に应答して、例外ハンドラが起動される。この例外ハンドラは、ステータスレジスタ91のリカバリモードビット92を「1」にセットし、かつRPC85の値をプログラムカウンタ4にセットする。これにより、このマイクロプロセッサは通常モードからリカバリモードになる。

【0300】RPC85には第1サイクルにおける命令I1~I4のアドレスが格納されているので、この第1サイクルから命令が再実行される。リカバリモードにおいては、実行制御回路11がTFレジスタ86を参照して投機的命令だけを実行する。したがって、第1サイクルにおいては、4つの命令のうち3つの投機的命令I2~I4だけが実行され、逐次的命令I1は実行されない。

【0301】ここで、命令I2が実行されるため、再び例外が生じる。このとき、命令I2のコミット条件TF[1]は将来「真」となるものか「偽」となるものかをFTFレジスタ88の第1エントリを参照してチェック

される。ここでは、FTFレジスタ88の第1エントリは「真」であるので、割込制御回路89によって生成された割込制御信号に应答して例外処理が行なわれる。このときTFレジスタ86の第1エントリはまだ「未定」であるが、この第1エントリは将来必ず「真」となるものであるからその例外処理が行なわれるのである。

【0302】ここで、もしも命令I3も例外を引起こしたとしても、この命令I3のコミット条件はTF[1]であるのに対し、FTFレジスタ88の第1エントリは「真」であるので、その例外処理が行なわれることはない。ここではFTFレジスタ88の第1エントリは「真」であるが、「未定」の場合は、通常モードと同様に、Hレベルの例外発生信号SEに应答してEP回路72における第5エントリのラッチ回路74が「1」にセットされる。

【0303】この例外処理は通常の例外処理と同様に、例外が発生したと同時に進行しているもので、その例外処理が終了すると、その例外を引起こした命令I2の次の命令C1、C2のアドレスがプログラムカウンタ4に格納され、これらの命令C1、C2から実行が再開される。このため、プログラムカウンタ4の値がEPC70の値と等しくなり、これにより比較回路90はHレベルの比較信号CMPを生成する。Hレベルの比較信号CMPに应答して、TFレジスタ86はFTFレジスタ88からの条件情報を格納し、さらにステータスレジスタ91のリカバリモードビット92は「0」にリセットされる。これにより、このマイクロプロセッサはリカバリモードから通常モードに戻る。

【0304】この第7実施例によれば、上記第6実施例に必要なリカバリコードとそのリカバリコードを呼出すためのジャンプテーブルとが必要とならないので、例外処理を行なうためにプログラムのバイナリサイズを特に大きくする必要はない。また、リカバリのための再実行は元のプログラムに従って行なわれるので、キャッシュミス、ページフォールトなどの例外が発生する確率は極めて低くなる。さらに、リカバリのための再実行が行なわれるときにおいても、このマイクロプロセッサは命令を投機的に実行することができるので、投機的実行が可能なこのマイクロプロセッサの機能をより有効に活用することができる。

【0305】

【発明の効果】請求項1に記載のコミット条件付き命令の処理方法によれば、コミット条件の成否を判定する前にその命令を投機的に実行することができるので、ALUなどの演算装置の使用率が高められ、命令の処理速度が向上するという効果を奏する。

【0306】請求項2に記載のコミット条件付き命令の処理方法によれば、上記請求項1の効果に加えて、投機的に実行した命令が例外を引起こした場合はその例外処理を保留し、コミット時にその例外処理を行なうので、

無駄な例外処理を行なうことがなく、命令の処理速度はさらに向上するという効果を奏する。

【0307】請求項3に記載のコミット条件付き命令の処理方法によれば、上記請求項1および請求項2の効果に加えて、既に処理した命令を再処理するので、例外が発生した場合でも常に正確な結果データが生成され、すべての命令を正確に処理することができる。

【0308】請求項4に記載のコミット条件付き命令の処理方法によれば、パイプライン制御によるある処理ステージにおいて、命令を実行して生成した結果データは確定的に保持されるべきか暫定的に保持されるべきかという処置情報が生成されるため、コミット条件の成否を判定する前に命令を投機的に実行することができ、ALUなどの演算装置の使用率がさらに高められ、命令の処理速度はさらに向上するという効果を奏する。

【0309】請求項5に記載のコミット条件付き命令の処理方法によれば、上記請求項4の効果に加えて、ある処理ステージにおいて上記処置情報が必要に応じて変更されるため、すべての命令を常に正確に処理することができる。

【0310】請求項6に記載のコミット条件付き命令の処理装置によれば、コミット条件の成否が判明する前にその命令を投機的に実行することができるので、ALUなどの演算装置の使用率が高められ、命令の処理速度が向上するという効果を奏する。

【0311】請求項7に記載のコミット条件付き命令の処理装置によれば、パイプライン制御によるある処理ステージにおいて、命令を実行して生成した結果データは確定的に保持されるべきか暫定的に保持されるべきかを示す処置情報が生成されるため、コミット条件の成否が判明する前に命令を投機的に実行でき、ALUなどの演算装置の使用率がさらに高められ、命令の処理速度はさらに向上するという効果を奏する。

【0312】請求項8に記載のコミット条件付き命令の処理装置によれば、上記請求項7の効果に加えて、そのある処理ステージにおいてコミット条件が判明した場合はその処置情報が変更されるので、命令を常に正確に処理することができる。

【0313】請求項9に記載のコミット条件付き命令の処理装置によれば、上記請求項6の効果に加えて、確定的に保持された結果データであっても暫定的に保持された結果データであっても、常に読出すことができるという効果を奏する。

【0314】請求項10に記載のコミット条件付き命令の処理装置によれば、請求項6の効果に加えて、フラグによって結果データが確定的に保持されているか暫定的に保持されているかが示されるので、レジスタファイルの容量はさほど大きくならない。

【0315】請求項11に記載のコミット条件付き命令の処理装置によれば、上記請求項6の効果に加えて、真

偽レジスタが条件情報を確定的または暫定的に保持することができるので、条件命令も投機的に実行することができる。

【0316】請求項12に記載のコミット条件付き命令の処理装置によれば、上記請求項6の効果に加えて、命令を投機的に実行したときに例外が発生した場合はそのコミット時に例外処理を行なうので、無駄な例外処理を行なうことはなく、命令の処理速度はさらに向上するという効果を奏する。

10 【0317】請求項13に記載のコミット条件付き命令の処理装置によれば、そのコミット時までの命令が再処理されるので、常に正しい結果データが生成される。したがって、常に命令を正確に処理することができるという効果を奏する。

【0318】請求項14に記載のコミット条件付き命令の処理装置によれば、上記請求項6、請求項12および請求項13の効果に加えて、例外処理を行なう場合はそのコミット時までの命令を逐次的な順序で処理するため、命令の処理速度はさらに向上するという効果を奏する。

20 【0319】請求項15に記載のコミット条件付き命令の処理装置によれば、上記請求項6の効果に加えて、将来真偽レジスタ手段にコミット条件が将来なるであろう条件情報を保持しておき、そのコミット時までの命令を再処理して再び例外が発生したときその将来真偽レジスタ手段を参照して必要に応じて例外処理を行なっている

30 【0320】請求項16に記載のコミット条件付き命令の処理装置によれば、上記請求項6および請求項15の効果に加えて、例外が発生した場合は投機的な命令だけが再処理されるので、例外を正確に処理できるという効果を奏する。

【0321】請求項17に記載のコミット条件付き命令の処理装置によれば、上記請求項14および請求項16の効果に加えて、例外の発生が一旦記憶されると、その内容が書換えられることはないので、その例外処理は確実に行なわれるという効果を奏する。

【図面の簡単な説明】

40 【図1】 この発明の第1実施例によるマイクロプロセッサの構成を示すブロック図である。

【図2】 図1に示したマイクロプロセッサのための命令コードの一例を示す説明図である。

【図3】 図1に示したマイクロプロセッサにおけるTFレジスタの構成を示す説明図である。

【図4】 図1に示したマイクロプロセッサのための命令コードにおけるコミット条件部の構成を示す説明図である。

50 【図5】 図1に示したマイクロプロセッサにおける実行制御回路の機能を説明するための概念図である。

【図6】 図1に示した実行制御回路の構成を示すブロック図である。

【図7】 図6に示した実行制御回路におけるコミット条件計算回路の構成を示すブロック図である。

【図8】 図7に示したコミット条件計算回路における比較回路の構成を示す回路図である。

【図9】 図1に示したマイクロプロセッサにおけるシーケンシャルレジスタファイル、シャドウレジスタファイルおよびコミット制御回路の構成を示すブロック図である。

【図10】 プログラムの一例を示すフローチャートである。

【図11】 図10に示したプログラムに従って図1に示したマイクロプロセッサのためのプログラムを示すプログラムリストである。

【図12】 この発明の第2実施例によるマイクロプロセッサの構成を示すブロック図である。

【図13】 図12に示したマイクロプロセッサの動作を示すタイムチャートである。

【図14】 図12に示したマイクロプロセッサにおける有効性計算回路の構成を示すブロック図である。

【図15】 図12に示したマイクロプロセッサにおける書込指定計算回路の構成を示すブロック図である。

【図16】 この発明の第3実施例によるマイクロプロセッサのためのプログラムを示すプログラムリストである。

【図17】 図16に示したプログラムに従って命令を処理するマイクロプロセッサにおけるコミット条件計算回路の構成を示すブロック図である。

【図18】 この発明の第4実施例によるマイクロプロセッサの構成を示すブロック図である。

【図19】 図18に示したマイクロプロセッサのためのプログラムを示すプログラムリストである。

【図20】 この発明の第5実施例によるマイクロプロセッサにおけるレジスタファイルの構成を示す説明図である。

【図21】 この発明の第6実施例によるマイクロプロ

セッサの構成を示すブロック図である。

【図22】 図21に示したマイクロプロセッサにおけるシーケンシャルレジスタファイル、シャドウレジスタファイル、コミット制御回路およびEP回路の構成を示すブロック図である。

【図23】 図21に示したマイクロプロセッサによる例外処理の動作を示すタイムチャートである。

【図24】 図21に示したマイクロプロセッサによって処理されるリカバリコードを示すプログラムリストである。

【図25】 図21に示したマイクロプロセッサにおいて、リカバリコードを呼出すためのプログラムリストである。

【図26】 図21に示したマイクロプロセッサにおけるEP回路に書換禁止の手段が必要な理由を説明するためのプログラムリストである。

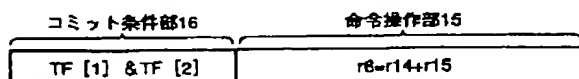
【図27】 この発明の第7実施例によるマイクロプロセッサの構成を示すブロック図である。

【図28】 従来のマイクロプロセッサの構成を示すブロック図である。

【符号の説明】

2A, 2B, 2C, 2D ALU、10, 86 TFレジスタ、11 実行制御回路、12 シーケンシャルレジスタファイル、13 シャドウレジスタファイル、14 コミット制御回路、17, 41, 53 コミット条件計算回路、20, 21 比較回路、38 コミット条件レジスタ、39 第1の有効性レジスタ、40 第1の書込指定レジスタ、42 有効性計算回路、43 書込指定計算回路、44 第2の有効性レジスタ、45 第2の書込指定レジスタ、52 レジスタファイル書込制御回路、60 シーケンシャルTFレジスタ、61 シャドウTFレジスタ、62 TFレジスタ用コミット制御回路、65 第1のレジスタファイル、66 第2のレジスタファイル、70 例外プログラムカウンタ (EPC)、72 EP回路、74 ラッチ回路、81 OR回路、87 TF計算回路、88 FTFレジスタ、89 割込み制御回路、F1~F32 フラグ。

【図2】



【図16】

```
always ? I1; TF [1] ? I2; /TF [1] ? I3; TF [2] ? I4;
always ? C1;
TF [1] ? C2;
TF [1] ? J1; /TF [1] ? J2; TF [2] ? J3;
```

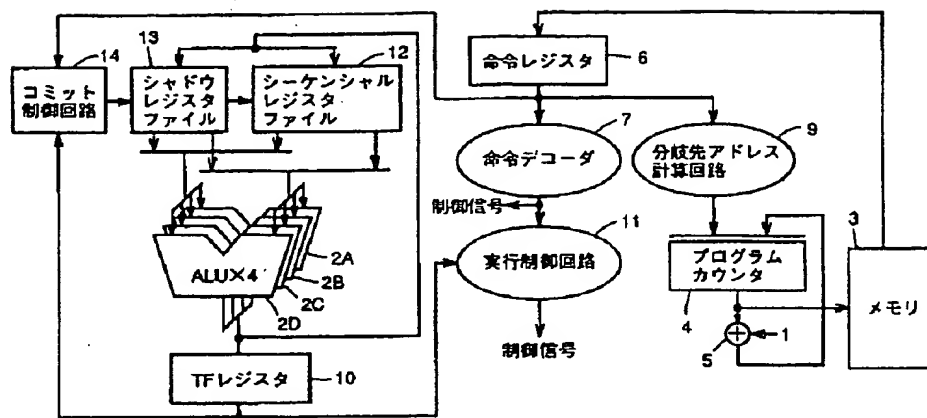
【図3】

TFレジスタ10					
第1エントリ	第2エントリ			第mエントリ	
有効性	値	有効性	値	有効性	値
Tv1	Td1	Tv2	Td2	Tvm	Tdm

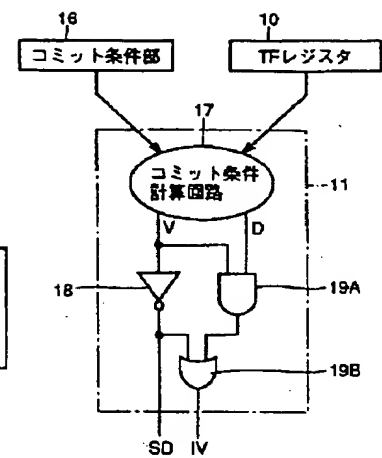
【図25】

```
recoveryCodeAddress=jumpTable[EPC];
goto recoveryCodeAddress;
```

【図1】



【図6】



【図4】

コミット条件部

第1エントリ		第2エントリ		第mエントリ	
有効性	値	有効性	値	有効性	値
Cv1	Cd1	Cv2	Cd2	Cvm	Cdm

【図5】



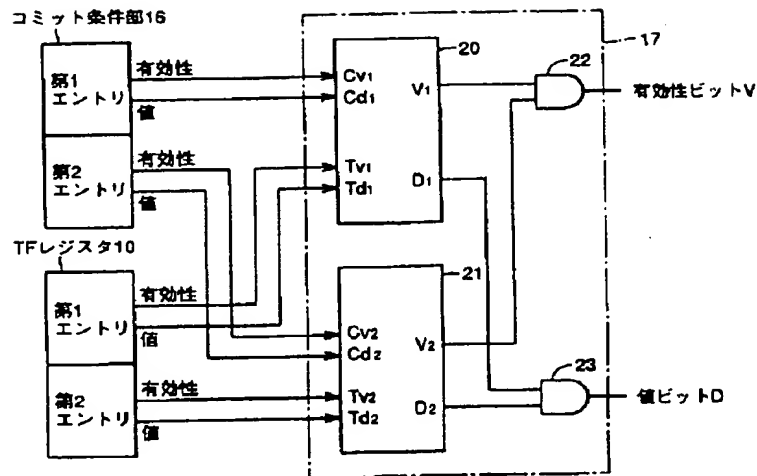
コミット条件部16

第1のエントリ	第2のエントリ	第3のエントリ	第4のエントリ
真	真	偽	Don't Care

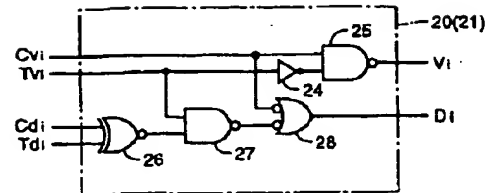
TFレジスタ10

	第1のエントリ	第2のエントリ	第3のエントリ	第4のエントリ	
(A)	真	真	偽	...	⇒ 限定的実行
(B)	未定	未定	未定	...	⇒ 暫定的実行
(C)	真	真	未定	...	⇒ 暫定的実行
	真	未定	未定	...	⇒ 暫定的実行

【図7】



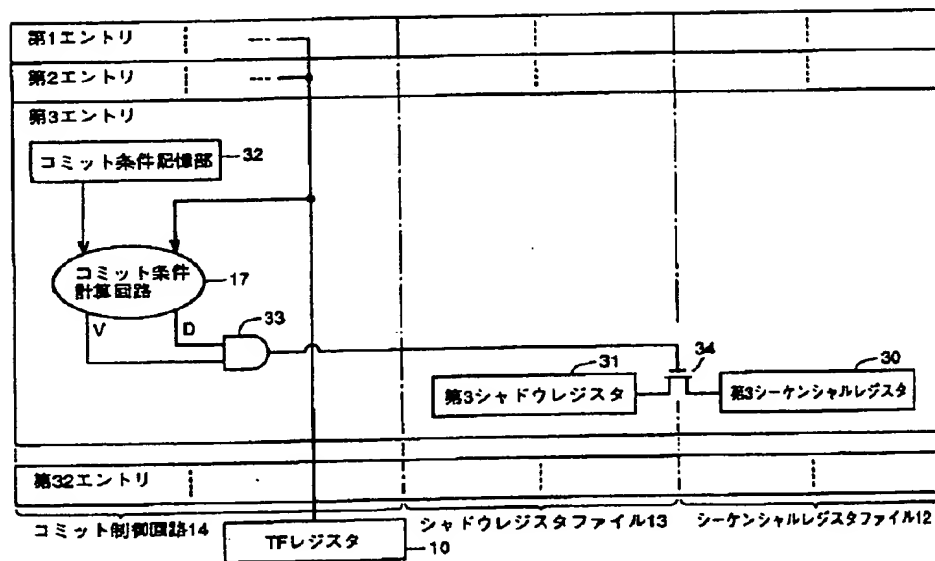
【図8】



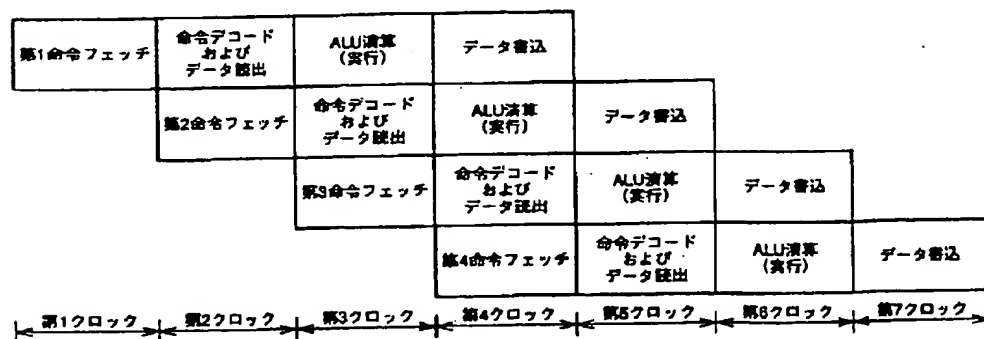
【図24】

always ? C1;
 TF[1] ? I2; /TF[1] ? I3;
 always ? C2;
 TF[1]&TF[2] ? I4;
 always ? jump(主プログラムにおける第3サイクルのアドレス)

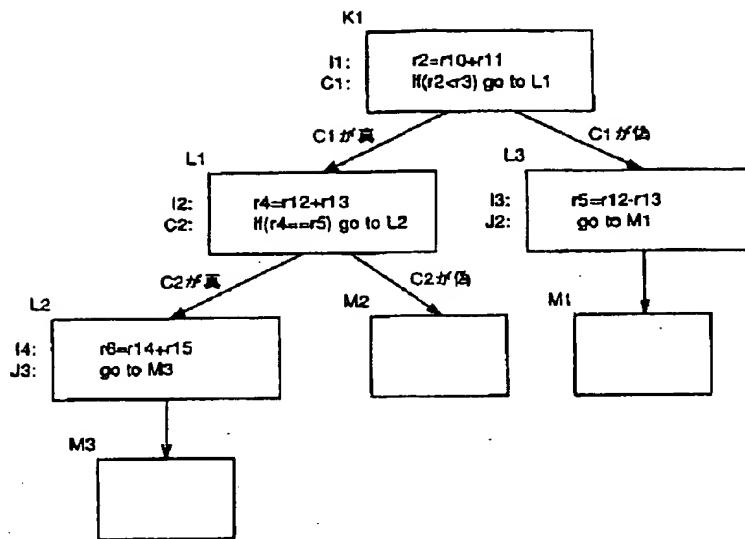
【図9】



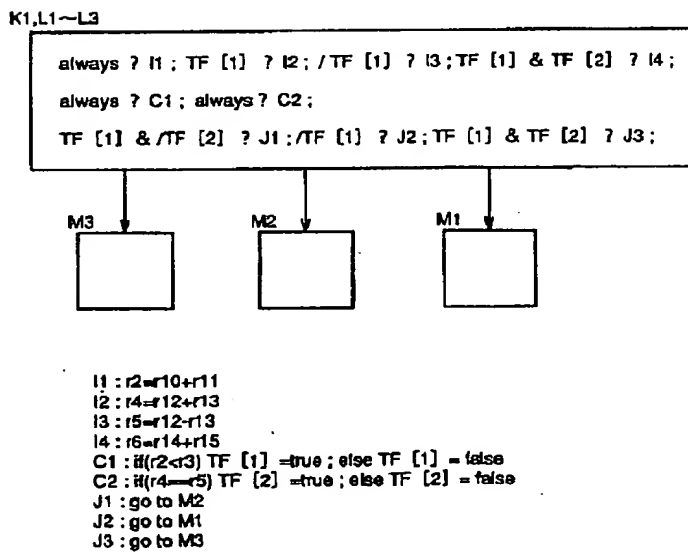
【図13】



【図10】



【図11】



【図19】

```

always ? I1 ;
/TF [1] ? I3 ;
always ? C1 ;
TF [1] ? J1 ;
TF [2] ? J3 ;
  
```

```

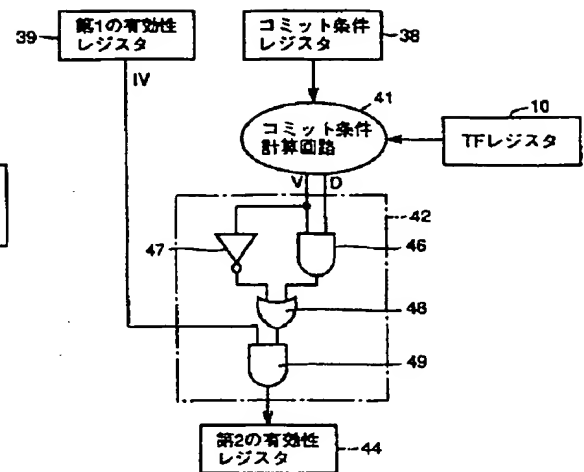
TF [1] ? I2 ;
/TF [1] ? C2 ;
TF [2] ? I4 ;
/TF [1] ? J2 ;
  
```

F1	RA1	RB1
F2	RA2	RB2
F3	RA3	RB3
...
F32	RA32	RB32

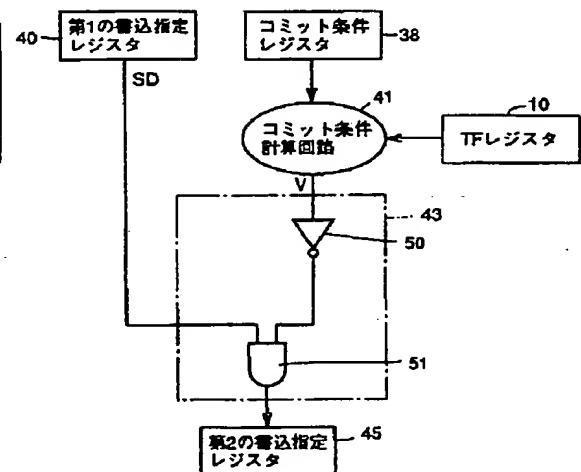
第1のレジスタファイル85

第2のレジスタファイル66

【図14】

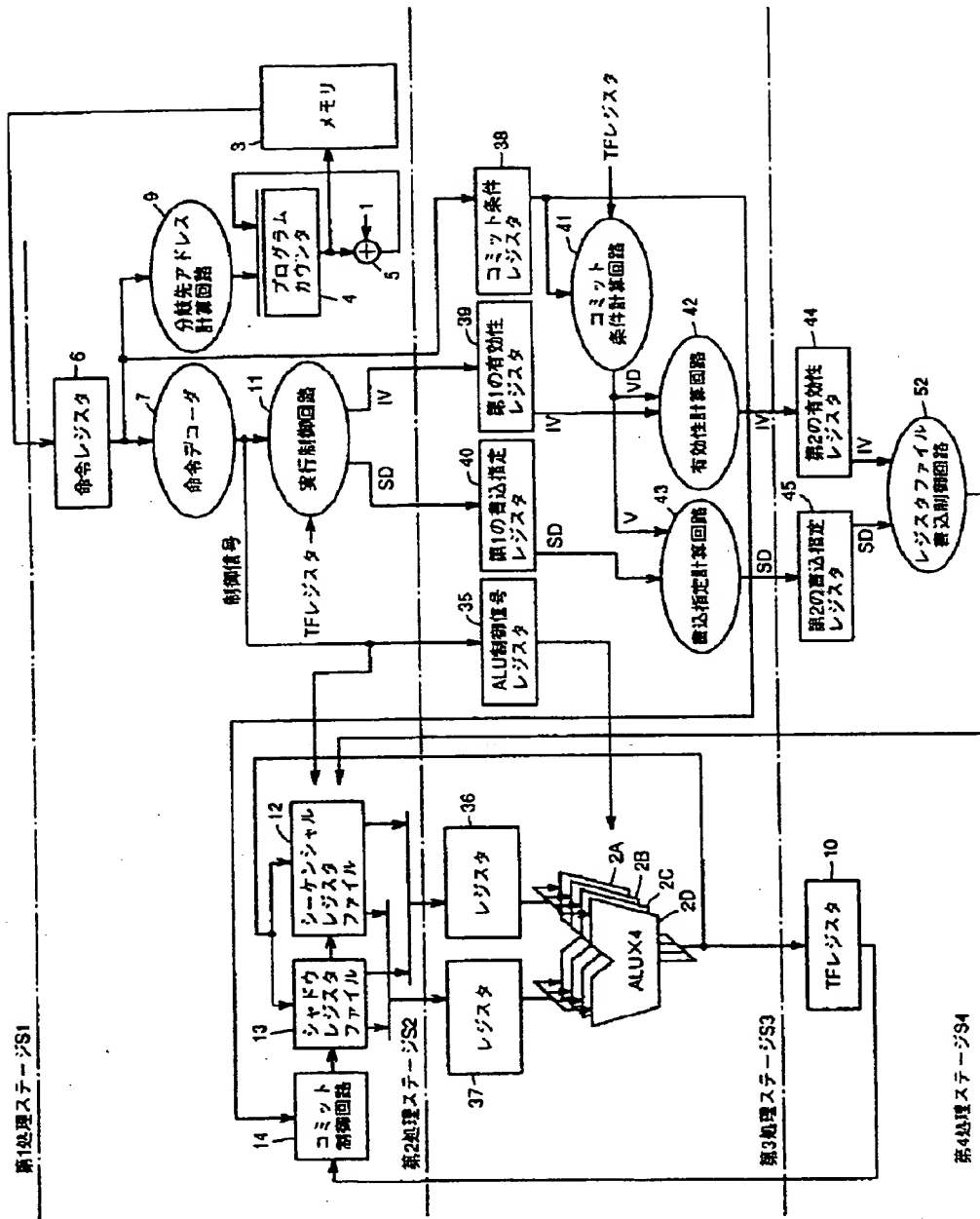


【図15】

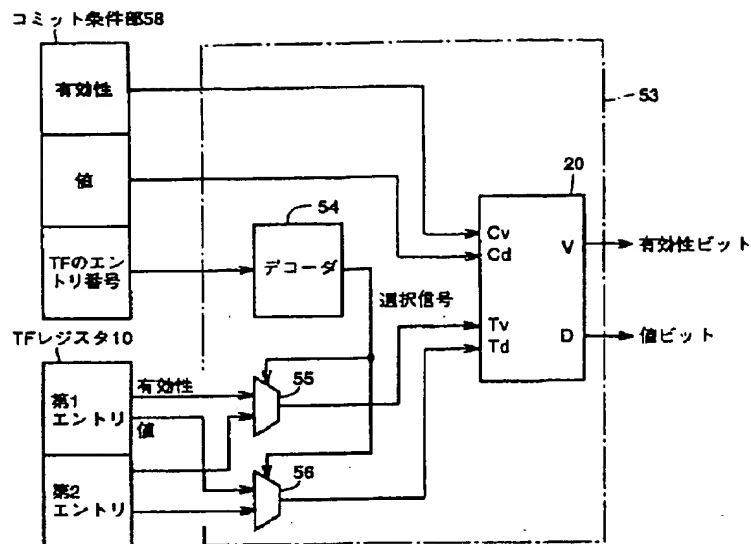


【図20】

【図12】



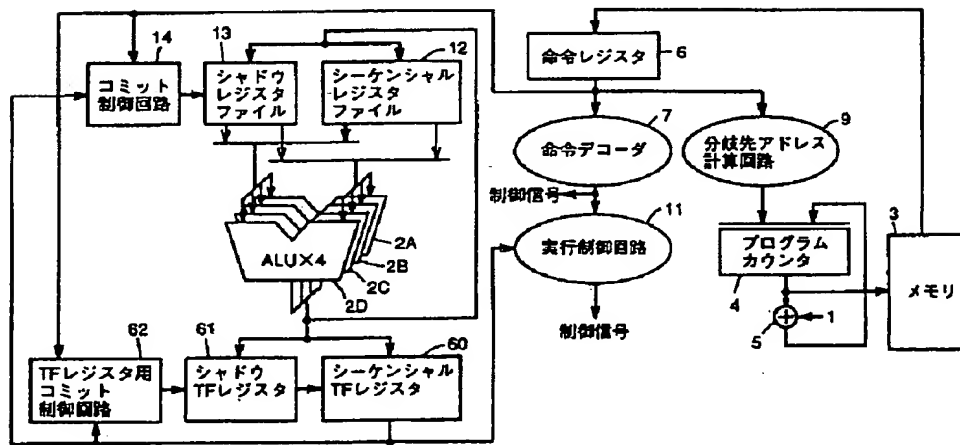
【図17】



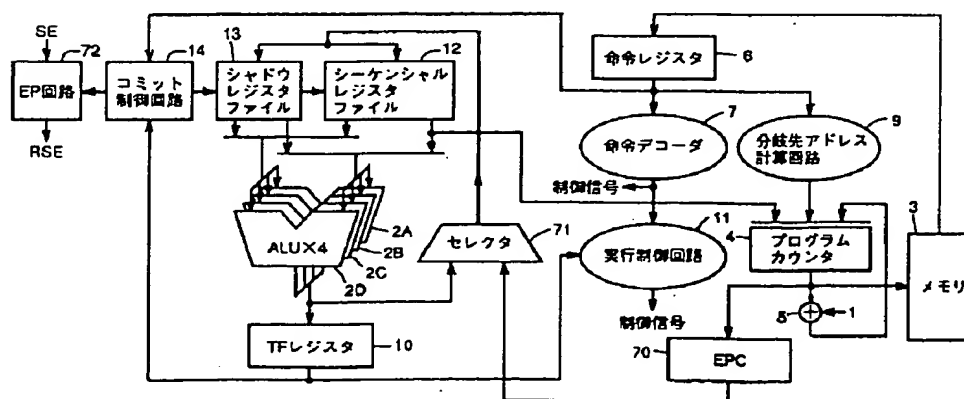
【図26】

110: TF[1] ? r1=r2+r3;
 111: TF[1] ? r1=r1+r2;
 112: always ? if (r3==r2) TF[1]=true, else TF[1]=false;

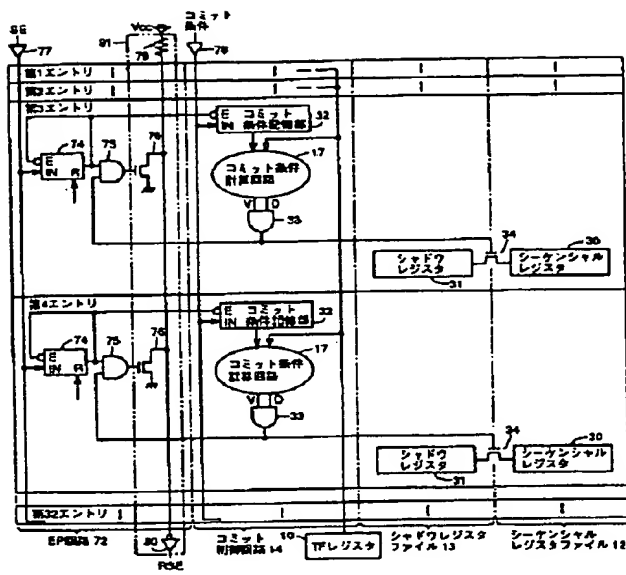
【図18】



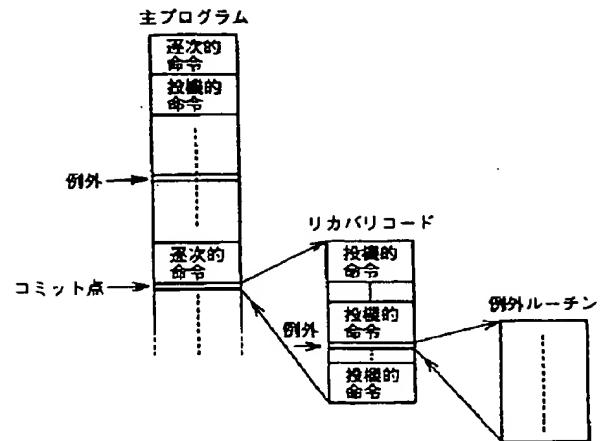
【図21】



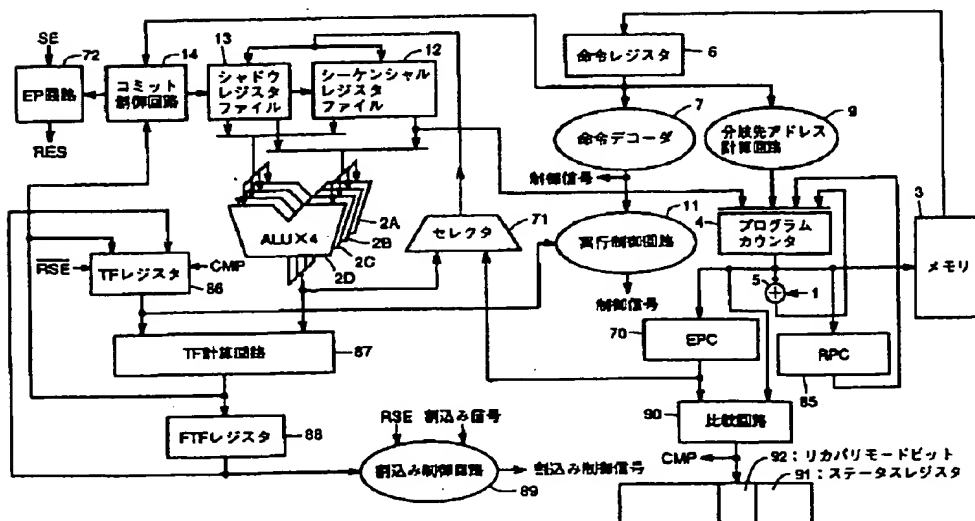
【図22】



【図23】



【図27】



【図28】

